



Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia  
Departamento de Informática

Dissertação de Mestrado em Engenharia Informática  
2008/2009

Geração automática de “playlists” de músicas semelhantes  
26332 Francisco Manuel Ramos Costa

Orientador  
Prof.<sup>a</sup>. Doutora Fernanda Maria Barquinha Tavares Vieira Barbosa

29 de Julho de 2009



Nº do aluno: 26332

Nome: Francisco Manuel Ramos Costa

Título da dissertação: Geração automática de “playlists” de músicas semelhantes

Palavras-Chave:

- Pesquisa por alcance
- Recuperação de informação musical
- Estruturas de dados métricas

Keywords:

- Similarity range query
- Music information retrieval
- Metric data structures



## Resumo

---

Os avanços tecnológicos em compressão de áudio e em redes de computadores possibilitaram um acréscimo significativo na disponibilidade da música. Lidar com colecções de músicas de grandes dimensões torna a pesquisa de músicas semelhantes num aspecto cada vez mais importante.

De forma a comparar músicas, são usadas medidas de distância para medir a semelhança entre duas músicas, com base em descritores extraídos do conteúdo musical. Infelizmente os métodos actuais para o cálculo de semelhança de músicas em grandes quantidades são bastante exigentes ao nível de processador e de memória. Em colecções de grandes dimensões torna-se impraticável a realização de pesquisas de músicas semelhantes de forma exaustiva. Por estes motivos é necessário o uso de técnicas que permitam melhorar a eficiência das pesquisas por semelhança em música.

De modo a tornar as pesquisas por semelhança mais eficientes, são usadas normalmente estruturas indexadas por distância, especialmente as estruturas de dados métricas.

Nesta dissertação pretende-se estudar a aplicabilidade e a eficiência de estruturas de dados métricas em aplicações de pesquisa por alcance de músicas semelhantes. É descrito um estudo dos aspectos relevantes na pesquisa aproximada de música, nomeadamente técnicas para representação de conteúdo musical, funções de cálculo de semelhança e estruturas de dados métricas. Os quais permitem efectuar o estudo, onde se pode concluir que estas estruturas são aplicáveis a este domínio da música e que possibilitam acelerar o processo de pesquisa por alcance.

---



## **Abstract**

---

Audio compression and computer networking technological advances have prompted a significant increase in the availability of music access to the general audience. Nevertheless, working with large-scale music collections turns the music similarity search into a vital aspect that has to be taken into account.

In order to compare music, distance measures are used to determine the similarity between two songs, based on specific descriptors extracted from the music content. Unfortunately, the available methods for calculating the similarity of songs in large scale are very demanding in terms of processor and memory. Furthermore, in large-scale collections it becomes impractical to perform searches of similar songs in an exhaustive way. For all these reasons it becomes necessary the use of specific and accurate techniques to improve the efficiency of similarity searches in music.

In order to achieve the higher efficiency possible in similarity searches, it is normally used a distance-based index structure, specifically the data metric structures.

This dissertation aims to study the applicability and efficiency of metric data structures in range query applications among similar songs. A study of relevant aspects in the research concerning music similarity search was carried out, including techniques for representation of music content, similarity functions and metric data structures. This allows the execution of the study, where it can be concluded that these structures are applicable to this music field and allow speeding up the search process.

---





## **Agradecimentos**

---

Em primeiro lugar, gostaria de expressar o meu agradecimento à minha orientadora, Prof. Doutora Fernanda Barbosa, por me ter aceite neste projecto de investigação, pela sua dedicação, apoio e encorajamento demonstrado durante a realização desta dissertação. Agradeço também ao Prof. Doutor Eduardo Lopes e ao Eng. Alfie Leong pelo seu suporte e disponibilidade neste trabalho. À Prof. Doutora Margarida Mamede e ao Prof. Doutor João Moura Pires pela sua aprovação na primeira parte deste projecto e pelas suas críticas e sugestões, sempre construtivas, que permitiram melhorá-lo.

Em segundo lugar, a todos os meus amigos e colegas, especialmente ao Pedro Chambel, Nuno Farruca e Ruben Jorge pelo apoio e companheirismo. Aos amigos Filipe, Luís, Silvino, Sylvia e Rute pela sua amizade e camaradagem.

Finalmente, gostaria de agradecer profundamente aos meus pais, Francisco e Aldina e à minha irmã Joana, pela enorme compreensão, apoio e motivação durante todos estes anos.

---



## Índice

---

1. Introdução .....	1
1.1 Motivação.....	2
1.2 Objectivo .....	3
1.3 Metodologia .....	3
1.4 Contribuições previstas .....	4
1.5 Estrutura .....	4
Parte I – Estado de arte .....	7
2. Aspectos introdutórios .....	9
2.1 Tipo de aplicações na área de recuperação de informação em música (MIR) .....	9
2.2 Conceitos relacionados com conteúdo musical.....	10
2.3 Formatos de ficheiros de música.....	11
2.3.1 Formato simbólico.....	11
2.3.2 Formato áudio .....	12
2.4 Conclusões .....	13
3. Representação do conteúdo musical.....	15
3.1 Melodia .....	15
3.1.1 Sequência de intervalos de notas.....	16
3.1.2 Sequência de pitch.....	17
3.1.3 Conjuntos de pontos ponderados de notas melódicas .....	18
3.2 Timbre .....	19
3.2.1 Mel-Frequency Cepstral Coefficients – MFCCs.....	20
3.3 Ritmo.....	21
3.3.1 Beat Spectrum .....	22
3.4 Conclusões e contributos.....	24
4. Semelhança de músicas .....	27
4.1 Distância de edição.....	27
4.2 $L_p$ Norm.....	29
4.3 Distância de transportação .....	30
4.3.1 Earth Mover Distance.....	31

4.3.2 Proportional transportation distance.....	32
4.4 Dynamic time warping (DTW) .....	36
4.5 Conclusões e contributos.....	36
5. Pesquisa aproximada em espaços métricos .....	39
5.1 “List of Clusters” – LC.....	42
5.1.1 Construção.....	43
5.1.2 Pesquisa.....	44
5.2 “Recursive List of Clusters” – RLC .....	45
5.2.1 Construção.....	46
5.2.2 Inserção .....	46
5.2.3 Remoção.....	47
5.2.4 Pesquisa.....	48
5.3 “Vantage Point Tree” – VPTree.....	50
5.3.1 Construção.....	50
5.3.2 Pesquisa.....	52
5.4 “Geometric Near-Neighbor Access Tree” – GNAT.....	54
5.4.1 Construção.....	55
5.4.2 Pesquisa.....	56
5.5 “Dynamic Spatial Approximation Tree” – DSAT .....	57
5.5.1 Construção.....	57
5.5.2 Inserção .....	58
5.5.3 Remoção.....	58
5.5.4 Pesquisa.....	59
5.6 “Hybrid Dynamic Spatial Approximation Tree” – H-DSAT .....	60
5.6.1 Construção.....	61
5.6.2 Inserção .....	61
5.6.3 Remoção.....	61
5.6.4 Pesquisa.....	61
5.7 “Linear Approximating Eliminating Search Algorithm” – LAESA .....	62
5.7.1 Construção.....	63
5.7.2 Pesquisa.....	64
5.8 Conclusões e contributos.....	65
Parte II – Implementação e Avaliação.....	67
6. Pacote de estruturas de dados métricas.....	69
6.1 Descrição das interfaces .....	70

6.2 Descrição das classes .....	70
6.2.1 LAESA .....	71
6.2.2 VPTree .....	71
6.2.3 DSAT, H-DSAT1 e H-DSAT2 .....	72
6.2.4 LC.....	72
6.2.5 GNAT.....	73
6.2.6 RLC .....	74
6.3 Conclusões .....	75
7. Espaços métricos .....	77
7.1 Timbre .....	78
7.1.1 Processamento da assinatura .....	78
7.1.2 Base de dados de músicas .....	80
7.1.3 Medida de semelhança .....	80
7.2 Melodia .....	82
7.2.1 Processamento da assinatura .....	82
7.2.2 Base de dados de músicas .....	84
7.2.3 Medida de semelhança .....	85
7.3 Conclusões .....	87
8. Pesquisa por alcance em estruturas de dados métricas.....	89
8.1 Metodologia .....	89
8.1.1 Conjunto de pesquisa .....	91
8.1.1 Raios de pesquisa .....	91
8.1.3 Parametrização das estruturas de dados métricas.....	92
8.2 Resultados da avaliação na dimensão timbre .....	94
8.3 Resultados da avaliação na dimensão melodia.....	97
8.4 Conclusão.....	99
9. Protótipo .....	101
9.1 Interface.....	101
9.2 Arquitectura.....	104
9.3 Processamento.....	105
9.4 Conclusões .....	106
10. Conclusões e trabalho futuro .....	107
10.1 Conclusões .....	107
10.2 Contribuições .....	108
10.3 Trabalho futuro.....	109

11. Bibliografia.....	111
Anexos .....	119
A1 Parametrização das estruturas de dados métricas .....	119
A1.1 Parametrização da LAESA.....	119
A1.2 Parametrização da DSAT, H-DSAT1 e H-DSAT2 .....	120
A1.3 Parametrização da LC .....	122
A1.4 Parametrização da GNAT .....	123
A1.5 Parametrização da RLC.....	124

## Índice de figuras

---

Figura 1. Pianoroll de Première Arabesque de Claude Debussy.....	12
Figura 2. Representação no domínio de tempo de Première Arabesque de Claude Debussy. ....	12
Figura 3. Melodias iguais com tempo e key diferentes. ....	16
Figura 4. Intervalo de pitch e cálculo do rácio IOI (LIR).....	17
Figura 5. Vectors de Pitch de duas melodias (antes da normalização).....	18
Figura 6. Extração de vectores de pitch. ....	18
Figura 7. Exemplo de uma música representada com o conjunto de pontos ponderados. ....	19
Figura 8. Exemplo de um espectrograma. ....	20
Figura 9. Diagrama do cálculo de MFCCs. ....	21
Figura 10. “Beat spectrum” de “Gould Prelude”.....	23
Figura 11. Matriz de auto-semelhança. ....	24
Figura 12. Matriz de auto-semelhança em tons de cizento.....	24
Figura 13. Ilustração do fluxo de pesos usando a EMD.....	34
Figura 14. Diagrama do cálculo da assinatura de uma música.....	35
Figura 15. Partição do espaço com base em 2 pivôs u8 e u11. ....	41
Figura 16. Partição do espaço com base em 4 agrupamentos, com centros em c1, c2, c3 e c4.....	42
Figura 17. Representação de uma “list of clusters”.....	43
Figura 18. Algoritmo de construção da “list of clusters”. ....	43
Figura 19. Algoritmo de pesquisa da “list of clusters”.....	44
Figura 20. Representação gráfica da “recursive list of clusters”.....	46
Figura 21. Algoritmo de inserção da “recursive list of clusters”.....	47
Figura 22. Algoritmo de remoção da “recursive list of clusters”. ....	48
Figura 23. Exemplos de situações de pesquisa.....	49
Figura 24. Exemplo de uma VPTree com dois pivôs.....	50
Figura 25. Algoritmo de construção da VPTree.....	51

Figura 26. Algoritmo de selecção de pivôs. ....	51
Figura 27. Algoritmo de pesquisa da VPTree. ....	52
Figura 28. Exemplo de uma VP <sup>S</sup> T com dois pivôs. ....	53
Figura 29. Representação de uma GNAT.....	54
Figura 30. Algoritmo de construção da GNAT. ....	55
Figura 31. Algoritmo de pesquisa da GNAT.....	56
Figura 32. Exemplo de uma DSAT com 5 elementos e aridade 2. ....	57
Figura 33. Algoritmo de inserção da DSAT.....	58
Figura 34. Algoritmo de pesquisa da DSAT. ....	60
Figura 35. Exemplo de uma LAESA com 15 elementos e pivôs <i>u8</i> e <i>u11</i> . ....	63
Figura 36. Algoritmo de construção da LAESA. ....	64
Figura 37. Diagrama de classes do pacote de estruturas de dados métricas.....	69
Figura 38. Implementação da estrutura LAESA. ....	71
Figura 39. Implementação da estrutura VPTree.....	71
Figura 40. Implementação das estruturas DSAT, H-DSAT1 e H-DSAT2.....	72
Figura 41. Implementação da estrutura LC. ....	73
Figura 42. Implementação da estrutura GNAT. ....	74
Figura 43. Implementação da estrutura RLC. ....	75
Figura 44. Processo de cálculo da assinatura de uma música com base nos MFCCs. ...	78
Figura 45. Exemplo de geração da assinatura da música “awake.wav”.....	79
Figura 46. Protótipo da similaridade na dimensão timbre.....	81
Figura 47. Processo de cálculo da assinatura de uma música com base nos intervalos melódicos.....	82
Figura 48. Exemplo de geração da assinatura da música “awake.mid”. ....	84
Figura 49. Padrão de cálculo da matriz de distância. ....	85
Figura 50. Protótipo da similaridade na dimensão melodia. ....	86
Figura 51. Método de avaliação. ....	90
Figura 52. Número de distâncias calculadas em cada espaço métrico da dimensão timbre.....	95
Figura 53. Número de distâncias calculadas no espaço métrico da dimensão melodia. ....	97
Figura 54. Interface do sistema.....	102
Figura 55. Exemplo da interface do sistema após a geração de uma “playlist”. ....	103
Figura 56. Diagrama de pacotes do Protótipo. ....	104
Figura 57. Parametrização da LAESA. ....	120



Figura 58. Parametrizações da DSAT, H-DSAT1 e H-DSAT2. ....	122
Figura 59. Parametrizações das duas versões da LC. ....	123
Figura 60. Parametrizações da GNAT. ....	124
Figura 61. Parametrizações da RLC. ....	125



## Índice de tabelas

---

Tabela I. Propriedades dos espaços métricos. ....	81
Tabela II. Propriedades do espaço métrico da melodia. ....	86
Tabela III. Número médio de músicas e percentagem da colecção de música obtida com a distância de Manhattan. ....	92
Tabela IV. Número médio de músicas e percentagem da colecção de música obtida com a distância euclidiana. ....	92
Tabela V. Número médio de músicas e percentagem da colecção de música obtida com a distância de edição em intervalos melódicos. ....	92
Tabela VI. Parâmetros da LAESA. ....	93
Tabela VII. Parâmetros da DSAT, H-DSAT1 e H-DSAT2. ....	93
Tabela VIII. Parâmetros das duas versões da LC. ....	93
Tabela IX. Parâmetros da GNAT. ....	94
Tabela X. Parâmetros da RLC. ....	94
Tabela XI. Percentagem dos ficheiros comparados da base de dados da dimensão timbre nas várias pesquisas por alcance. ....	96
Tabela XII. Percentagem dos ficheiros comparados da base de dados da dimensão melodia nas várias pesquisas por alcance. ....	98



## 1. Introdução

A distribuição de música mudou drasticamente ao longo da última década, fundamentalmente devido aos avanços tecnológicos na área de redes de computadores e das tecnologias dos algoritmos de compressão.

Actualmente uma colecção de música já não é considerada como um conjunto de “compact disks” (cds) armazenados em caixas. Presentemente uma colecção de música é vista como um conjunto, normalmente, de milhares de ficheiros digitais de áudio comprimidos num disco rígido de um computador ou em leitores portáteis.

Algoritmos de compressão de áudio, como por exemplo o MP3 (MPEG-1/2 Audio Layer 3)[56], inovaram o mundo da música. Estes vieram alterar a escala do armazenamento de músicas, possibilitando que milhares de ficheiros de áudio possam ser guardados num único computador pessoal preservando a qualidade sonora.

A Internet de certa forma também afectou o mundo da música. As melhorias na largura de banda, redes “peer-to-peer” e facilidades de acesso a este meio, possibilitaram uma distribuição e transmissão de ficheiros de áudio rápida e fiável. Lojas de venda de música online como o “iTunes Musical Store” e o “Napster” constituem um exemplo da integração destes avanços tecnológicos. A introdução destas novas facilidades culminou, por um lado, numa melhoria do bem-estar individual, uma vez que a cada indivíduo é dada a oportunidade de coleccionar uma vasta gama de ficheiros de música, a um ritmo célere; e por outro lado, numa vertente mais tecnológica, levaram à substituição do álbum, como unidade padrão de música, pela canção.

Toda esta evolução levou ao aparecimento de uma área de investigação, recuperação de informação em música (“Music Information Retrieval” - MIR), que contribuiu para a criação de software visando a gestão de música. A partir deste é possível criar e reproduzir música bem como gerir as diferentes canções armazenadas no computador. Uma das funcionalidades inerentes a estas aplicações é a criação de “playlists”. A criação de uma “playlist” pode ser manual ou automática. Por exemplo, criar uma “playlist” das canções favoritas poderia ser manual. No entanto, criar

“playlists” para agrupar canções de um dado género, de um dado autor ou semelhantes a uma dada canção, são exemplos de criações automáticas. A pesquisa na base de dados de canções pode ser feita por atributos associados aos elementos da base de dados, como por exemplo o seu autor, ou pode ser feita pelo conteúdo musical da canção, como por exemplo a semelhança entre músicas. A criação de uma “playlist” com base em elementos associados a um ficheiro de áudio depende da veracidade dos mesmos. Um exemplo desta situação são as ID3Tags dos ficheiros MP3. No entanto, é necessária a inserção manual das “tags” e em muitos dos casos tal não acontece ou possivelmente alguma informação pode ser incorrecta podendo levar ao aparecimento de diferenças significativas entre os ficheiros da “playlist”. Utilizando o conteúdo musical das músicas, tais erros são minimizados e possibilita-se aos utilizadores a descoberta de novas músicas com base em músicas que fazem parte do seu agrado.

Os trabalhos desenvolvidos para a procura de músicas semelhantes (pesquisa aproximada) baseiam-se em funções de semelhança entre músicas. Actualmente, está disponível uma vasta gama de medidas com este intuito, que se baseiam em diferentes aspectos intrínsecos às próprias músicas. Quando esta função de semelhança é métrica, o conjunto de todas as possíveis canções juntamente com a função chama-se espaço métrico. Na situação em que a função é métrica, é possível usar estruturas de dados métricas para organizar a base de dados (coleção de música), de forma que numa pesquisa seja possível não percorrer a totalidade do espaço de procura evitando o cálculo de semelhança entre uma dada música e todas as músicas da base de dados. Nesta situação é possível pesquisar as  $k$  músicas mais semelhantes a uma determinada música, ou então as músicas mais semelhantes de acordo com uma dada tolerância.

## **1.1 Motivação**

Actualmente, devido à tecnologia disponível, é possível a um único indivíduo possuir uma vasta coleção de ficheiros de áudio, na ordem de milhares de ficheiros de música. Gerar automaticamente “playlists” de músicas semelhantes a uma dada música com complexidade linear sobre o número de músicas é um processo exaustivo, uma vez que é necessário calcular a distância entre cada uma das músicas na base de dados e a música dada. Pelo que técnicas eficientes para pesquisa aproximada são essenciais.

Normalmente quando são feitas pesquisas baseadas em distâncias, são utilizadas estruturas de dados indexadas por distância. Tendo uma métrica para o cálculo de

semelhança entre músicas, é possível ter um espaço métrico que contém as músicas da base de dados. Assim sendo, é possível usar estruturas de dados métricas para a pesquisa de músicas semelhantes. Isto possibilita na maioria dos casos que não seja necessário percorrer a totalidade do espaço de pesquisa e torna a geração automática de uma “playlist” de músicas semelhantes mais eficiente.

## **1.2 Objectivo**

Nesta dissertação pretende-se estudar a aplicabilidade e a eficiência de estruturas de dados métricas em aplicações de pesquisa por alcance de músicas semelhantes. Para tal pretende-se implementar um protótipo que, com base no conteúdo extraído de músicas, gere automaticamente uma lista de músicas semelhantes, uma “playlist”, de forma eficiente, utilizando estruturas de dados métricas.

Este estudo envolve duas bases de dados de músicas, onde cada uma delas é definida sobre diferentes espaços métricos em duas dimensões da música. As pesquisas por alcance realizadas foram efectuadas em nove estruturas de dados métricas. A avaliação da pesquisa por alcance é realizada com base no número de distâncias calculadas.

## **1.3 Metodologia**

Com o objectivo de estudar a aplicabilidade e a eficiência de estruturas de dados métricas em aplicações de pesquisa por alcance de músicas semelhantes, nesta dissertação são elaborados dois estudos: o estudo da aplicabilidade e o estudo da eficiência. De forma a estudar a aplicabilidade das estruturas de dados métricas em aplicações de semelhança de música, é necessário:

- Estudar técnicas para a representação de conteúdo musical;
- Estudar medidas de semelhança do conteúdo musical;
- Implementar diferentes espaços métricos;
- Verificar a correcção dos espaços métricos, isto é, se estes exprimem a similaridade de músicas correctamente.

Este estudo foi realizado com base nas aplicações de recuperação de informação em música, principalmente publicados na conferência mais importante na área, ISMIR - “International Society for Music Information Retrieval”[52].

Quanto ao estudo da eficiência das estruturas de dados métricas em aplicações de semelhança de música, nomeadamente na pesquisa por alcance, é necessário:

- Estudar diversas estruturas de dados métricas;
- Implementar as diferentes estruturas de dados métricas;
- Avaliar a eficiência da pesquisa por alcance das estruturas de dados em diferentes espaços métricos sobre a respectiva base de dados.

#### **1.4 Contribuições previstas**

Esta dissertação pretende contribuir para tornar mais eficiente o processo de pesquisa de músicas semelhantes, principalmente no campo da eficiência temporal. Para além disso, é previsto que seja um contributo para a área de recuperação de informação em música (MIR), uma vez que será feito:

- Um estudo de vários espaços métricos existentes, no qual são descritas várias possíveis representações e funções de semelhança de música;
- Uma avaliação das diferentes estruturas de dados métricas em pesquisa por alcance no domínio da similaridade de músicas.

Para além das contribuições relativas a esta avaliação da aplicabilidade e eficiência das estruturas de dados métricas em aplicações de pesquisa de música semelhante, neste trabalho pretende-se implementar um protótipo que possibilite a geração de uma “playlist” com base numa música de pesquisa e num valor de alcance.

#### **1.5 Estrutura**

Este documento encontra-se organizado em duas partes compostas por dez capítulos. Neste primeiro capítulo, é feita uma apresentação introdutória ao tema, referindo o seu contexto, objectivo, motivação, metodologia e possíveis contribuições. Depois do capítulo 1, a primeira parte descreve o estado de arte dos aspectos/técnicas dos principais temas de investigação abordados nesta dissertação. Esta primeira parte é composta por quatro capítulos:

- Aspectos introdutórios: apresenta uma abordagem introdutória ao tema de recuperação de informação em música (MIR), indicando alguns conceitos e termos utilizados quando se fala de conteúdo musical;



- Representação do conteúdo musical: apresenta as dimensões mais relevantes da música e técnicas para a sua representação;
- Medidas de semelhança: apresenta as várias funções de comparação de música com base nas representações estudadas no capítulo 3;
- Pesquisa aproximada em espaços métricos: apresenta os conceitos referentes à pesquisa por semelhança em espaços métricos e descreve as principais estruturas de dados métricas;

A segunda parte descreve a implementação e as avaliações efectuadas nesta dissertação, sendo esta composta também por quatro capítulos:

- Pacote de estruturas de dados métricas: apresenta a descrição da implementação do pacote de estruturas de dados métricas;
- Espaços métricos: apresenta a descrição dos espaços métricos implementados e a sua avaliação informal relativa à aplicabilidade das estruturas de dados métricas;
- Pesquisa por alcance: apresenta a avaliação da eficiência das estruturas de dados métricas nos espaços métricos apresentados no capítulo 7;
- Protótipo: apresenta o protótipo desenvolvido nesta dissertação que possibilita a geração de “playlists” de músicas semelhantes, nomeadamente a descrição da sua interface e arquitectura;

No último capítulo apresenta-se as conclusões e possíveis direcções de investigação futura.

No final do documento é apresentado o anexo A1 - Parametrização das estruturas de dados métricas – que descreve o processo para determinar as parametrizações das estruturas de dados métricas.



## **Parte I – Estado de arte**



## **2. Aspectos introdutórios**

Um dos aspectos fulcrais das aplicações que operam ao nível do conteúdo musical é, sem dúvida, a representação da música com base no seu conteúdo. Todas estas aplicações tratam de consultar e/ou classificar colecções de músicas com base no conteúdo musical, estando por isso incluídas na área de recuperação de informação em música (MIR).

Este capítulo apresenta alguns aspectos relacionados com a música, nomeadamente tipos de aplicações na área de MIR, formatos de ficheiros de música e conceitos relacionados com o conteúdo musical. Assim sendo, na primeira secção são apresentados os tipos de aplicações existentes. Na segunda secção são apresentados um conjunto de conceitos musicais importantes, usados ao longo desta dissertação. Por último, é feita uma breve descrição dos formatos mais utilizados de ficheiros de música.

### **2.1 Tipo de aplicações na área de recuperação de informação em música (MIR)**

A maior parte dos trabalhos na área de MIR dizem respeito à classificação de músicas [54] e à consulta de músicas semelhantes a uma dada voz [3]. No entanto, foi encontrado um trabalho recente que faz consulta de músicas semelhantes com estruturas de dados métricas [1], o qual é também o objectivo desta dissertação.

Ainda que a maior parte dos trabalhos sejam dos tipos mencionados anteriormente, na área MIR é também possível classificar as aplicações em [55]:

- Classificação de género – Útil para organizar grandes colecções de música automaticamente.
- Identificação de compositor – Permite identificar a autoria de músicas entre vários artistas.
- Previsão de sucessos – Permite prever o sucesso comercial de músicas de modo a alocar eficientemente recursos financeiros. Com base na semelhança entre músicas semelhantes.
- Sistemas de recomendações – Permite aos utilizadores encontrarem músicas desconhecidas que podem ser do seu agrado.

- Identificação de duplicados – Permite a identificação de músicas com nome ou designações diferentes mas com o mesmo conteúdo musical.
- Geração automática de “playlists” – Permite a criação de uma “playlist” baseando-se em vários critérios, entre os quais, a utilização de uma dada música como semente para gerar um conjunto de músicas (“playlist”) semelhantes. Esta aplicação consiste num dos objectivos desta dissertação, tentando observar resultados anteriormente obtidos que indiciam que utilizando n músicas próximas à música dada se obtêm resultados satisfatórios [40], criando-se assim a “playlist” pretendida.

## 2.2 Conceitos relacionados com conteúdo musical

Existem alguns conceitos associados ao conteúdo musical que podem não ser familiares às pessoas que não possuem um estudo na área de recuperação de informação em música. De entre estes, os principais conceitos básicos no qual este trabalho assenta são:

- Attack time ou Onset – Ponto métrico onde ocorre um som. De uma forma geral, designa o ponto de início de um som no eixo do tempo.
- Intensidade – Volume sonoro. Em acústica e música refere-se à percepção da amplitude da onda sonora.
- Interonset interval – O intervalo “interonset interval”, normalmente abreviado por “IOI”, designa o tempo entre os “onset” de notas sucessivas. Este intervalo não inclui informação relativa à duração da nota musical.
- Tonalidade – É a escala das notas musicais, por exemplo a escala de dó ou de ré.
- Key signature – É uma indicação da tonalidade de uma música, normalmente mencionada no início de uma pauta.
- Loudness – É uma medida subjectiva, passível de variação entre indivíduos, que descreve a força/intensidade da percepção de um som.
- Offset time - Ponto métrico onde acaba um som. Geralmente, designa o ponto final de um som no eixo do tempo.
- Pitch – Representa a frequência fundamental percebida num som. De uma forma geral, é vulgarmente designado como as notas musicais, exemplo: dó, ré... .

- Tempo – é a velocidade à qual uma música é reproduzida, normalmente medido em “beats per minute”.
- Timbre - Qualidade sonora de um instrumento ou voz (na secção 3.2 será abordado de forma mais elucidativa este conceito).
- Time signature – Indicação da unidade da divisão de uma secção musical no espaço temporal. Especifica o número de “beats” em cada medida, normalmente representada como um número fraccionário, onde o numerador indica o número de “beats” e o denominador a duração da nota que representa uma “beat” (a unidade “beat”).

## **2.3 Formatos de ficheiros de música**

Ao longo deste trabalho o termo música é utilizado no seu sentido de uma forma geral, representando uma canção independentemente do formato de ficheiro em que se encontra armazenada. No entanto é importante esclarecer que existem diferentes formatos: o formato simbólico e o formato de áudio. Este aspecto é extremamente importante uma vez que a extracção de características da música depende do formato em que esta se encontra. Um exemplo desta especificidade é o facto de com base em ficheiros simbólicos não ser possível inferir acerca do timbre da música [15].

### **2.3.1 Formato simbólico**

O formato simbólico permite representar a música como uma organização estruturada de símbolos que correspondem a eventos acústicos, a partir dos quais é possível a sua produção. Este formato necessita de aplicações que permitam editar e visualizar a música. Por exemplo, na Figura 1 (retirada de [15]) está patente uma aplicação para a representação visual de uma música, denominado “pianoroll”, em que as notas aparecem como linhas horizontais, sendo o seu valor em y o pitch (nota). Actualmente, na área de MIR são utilizados vários formatos simbólicos como Midi [59], GUIDO [30], MuseData [31], MusicXML [32] e Lilypond [33].

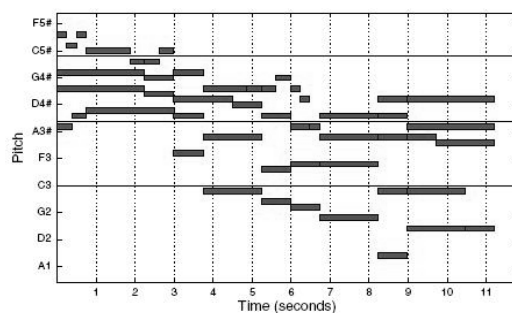


Figura 1. Pianoroll de Première Arabesque de Claude Debussy.

É de notar que o formato Midi - “Musical Instrument Digital Interface”, é considerado um compromisso entre o formato simbólico e de áudio [15]. No entanto este formato consiste numa representação simbólica da música, uma lista de mensagens (consultar [34] para uma descrição completa das mensagens) que, ao serem interpretadas por um sintetizador (teclado ou uma placa de áudio), reproduzem o som de instrumentos musicais com variadas características. Este tipo de formato simbólico ainda é amplamente utilizado em aplicações de música, embora se verifique um declínio face ao formato de áudio.

### 2.3.2 Formato áudio

Um ficheiro em formato de áudio consiste na representação de gravações digitais de trabalhos musicais, onde a aquisição digital é baseada em amostragem de sinal e quantificação dos valores de amostra. A maioria dos formatos áudio não comprimidos são baseados na representação “pulse code modulation” - PCM, onde cada amostra é representada como uma frequência com uma dada amplitude no domínio de tempo, Figura 2 (retirada de [15]).

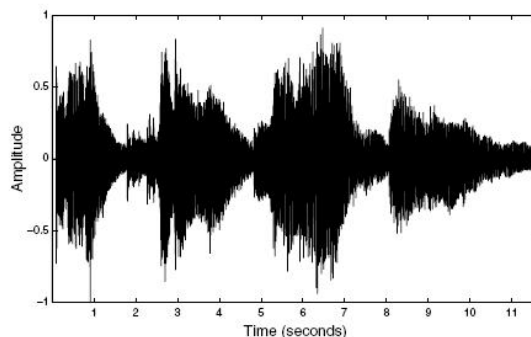


Figura 2. Representação no domínio de tempo de Première Arabesque de Claude Debussy.



Os formatos de áudio Wave e AIFF são os formatos standard para os sistemas da Microsoft/IBM e Macintosh, respectivamente, onde a “stream” de áudio em formato PCM é armazenada em segmentos. Normalmente, estes formatos apresentam elevadas dimensões quando comparados com outros formatos, como por exemplo o MP3, AAC e WMA.

Devido à restrição do espaço de armazenamento requerido pelos formatos baseados em PCM, foram desenvolvidos formatos de áudio comprimido – MP3, AAC e WMA – onde o sinal de áudio é armazenado de forma comprimida no domínio de frequências. A compressão é realizada pela remoção das frequências preceptivamente inaudíveis pelo ouvido humano mantendo praticamente intacta a qualidade sonora. Em [35] o leitor mais interessado pode consultar as técnicas usadas no MP3 e AAC, e em [36] e [37] dois métodos de semelhança usando directamente o formato comprimido MP3. Relativamente ao formato WMA, verifica-se que é um formato proprietário da Microsoft, sem documentação disponível, sendo esta uma das explicações possíveis para a sua débil difusão e divulgação, e por conseguinte de poucos trabalhos com este formato na área de MIR.

O formato de áudio MPEG-7 [38] consiste numa evolução no âmbito de semelhança de músicas, na medida em que este standard considera também a codificação de descritores baseados no conteúdo para tarefas da área de MIR.

## **2.4 Conclusões**

Neste capítulo foram apresentados diferentes tipos de aplicações que estão em desenvolvimento na área de MIR, nomeadamente a geração automática de “playlists” de música semelhante.

São definidos alguns conceitos associados ao conteúdo musical que serão utilizados ao longo desta dissertação.

Por último foi apresentada uma explicação relativa aos formatos de música onde se observa que o formato escolhido influencia de forma significativa a capacidade de extracção de descritores da música.



### **3. Representação do conteúdo musical**

O primeiro passo para aquisição de informação da música baseia-se no processamento automático de ficheiros de música, com a finalidade de extrair descritores relevantes do seu conteúdo. De acordo com a teoria e análise de música, esta última é caracterizada por um conjunto de dimensões [15], que embora relacionadas, podem ser usadas individualmente para descrever o conteúdo musical. São indicadas como dimensões da música: o timbre, o ritmo, a melodia, a acústica, a orquestração e a sua estrutura [15][23]. No entanto, com base no estudo realizado na área de MIR e publicado na ISMIR são identificadas como as dimensões mais relevantes: o timbre, o ritmo e a melodia.

Nas secções seguintes são apresentadas as diferentes formas de representação de conteúdo musical baseadas nas dimensões melodia, timbre e ritmo. Os descritores associados a cada dimensão apresentada são aqueles que possuem um lugar de destaque na área da MIR<sup>1</sup>.

#### **3.1 Melodia**

Melodia, “tune”, ou voz, consiste numa sucessão linear de tons musicais que é percebida como uma única entidade. No seu sentido mais simples, a melodia é uma sequência de pitches e durações, transmitida por um instrumento ou pela voz de um artista.

A melodia por si só constitui uma característica multidimensional, uma vez que é possível existir diferentes melodias em simultâneo (polifonia).

A extracção da melodia da música é um processo muito difícil. Primeiro que tudo, esta extracção é dependente do formato do ficheiro da música, se é monofónica ou polifónica e do género da música.

Em formatos simbólicos, como por exemplo o Midi, a extracção da melodia é directa no caso de música monofónica, onde qualquer símbolo representa o conteúdo melódico. No entanto, para a música polifónica, a extracção é mais complexa, uma vez

---

<sup>1</sup> Foram utilizados recentemente em trabalhos de investigação e publicados em ISMIR.

que existem várias linhas melódicas. Em particular, na música pop e rock é assumido que existe uma única linha melódica – a voz [15]. Para os ficheiros áudio são usadas técnicas de transcrição do pitch, onde a frequência fundamental de cada fonte de som é processada para cada frame [15][14].

Uma melodia pode ser representada de diferentes formas. Nas secções seguintes são apresentadas algumas que foram utilizadas em pesquisas recentes.

### 3.1.1 Sequência de intervalos de notas

Um intervalo de notas é representado como um par de valores, o pitch relativo entre duas notas consecutivas e o logaritmo do rácio entre a duração da nota corrente e da nota seguinte (LIR)[2]. Esta duração corresponde ao “interonset interval” (IOI). A codificação em intervalos de notas apresenta as vantagens de que melodias com tempo e “key signature” diferentes possuem um elevado nível de semelhança, ou seja, invariantes relativamente ao tempo e transposição. Por exemplo a Figura 3 (retirada de [2]), mostra um exemplo de duas melodias  $S_1$  e  $S_2$  com velocidades diferentes, sendo necessário usar uma representação melódica que caracterize bem este aspecto.

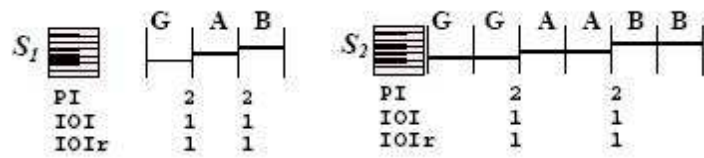


Figura 3. Melodias iguais com tempo e key diferentes.

Em trabalhos como [1] e [2], a melodia é representada como uma sequência de intervalos de notas, pelo que é necessário calcular para cada nota  $N_i$  as equações abaixo descritas.

Pitch relativo:

$$P_{rel}(N_i) = P_{abs}(N_i) - P_{abs}(N_{i-1})$$

$$P_{rel}(N_1) = 0$$

“Interonset interval”:

$$IOI(N_i) = t_{onset}(N_{i+1}) - t_{onset}(N_i)$$

$$IOI(N_n) = t_{offset}(N_n) - t_{onset}(N_n)$$

Rácio do “interonset interval”:

$$IOIR(N_i) = \frac{IOI(N_i)}{IOI(N_{i-1})}$$

$$IOIR(N_1) = 1$$

Logaritmo do rácio do “interonset interval”:

$$\text{LogIOIR}(N_i) = \log(\text{IOIR}(N_i))$$

Na Figura 4 (retirada de [3]), é mostrado para um excerto de música, os valores associados a cada uma das medidas necessárias para a extracção da melodia. As equações mostram como os valores dos intervalos são obtidos.

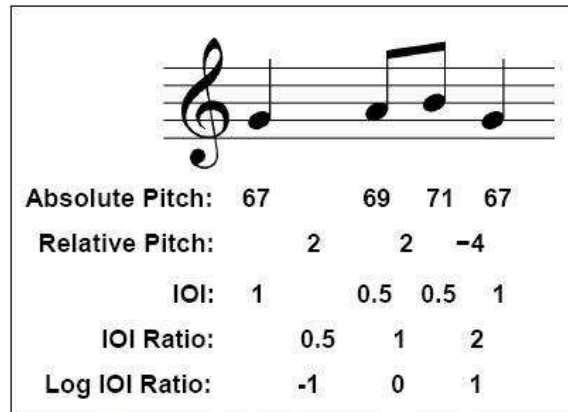


Figura 4. Intervalo de pitch e cálculo do rácio IOI (LIR).

Com base na Figura 4, a sequência de intervalos de notas associada à melodia seria “< 2, -1 > < 2, 0 > < -4, 1 >”.

### 3.1.2 Sequência de pitch

De uma forma geral, a melodia, é vista como uma sequência de notas, onde cada nota é representada por um triplo  $(p_i, b_i, e_i)$ . O elemento  $p_i$  corresponde ao pitch,  $b_i$  ao tempo de início (“onset time”) e  $e_i$  ao tempo de fim (“offset time”) em segundos. Em trabalhos como [4][14], a melodia é representada como um vector de pitch  $\langle p_1, \dots, p_d \rangle$ , sendo  $d$  o número de notas utilizadas.

Em [4][14], para efectuar a extracção do vector de pitch, primeiro eliminou-se os silêncios entre notas consecutivas estendendo o tempo de fim (“offset time”) de uma nota até ao tempo de início (“onset time”) da nota seguinte (este intervalo é designado por IOI), i. e.,  $e_i = b_i + 1$  para  $i = 1$  até  $d - 1$ . Posteriormente, procedeu-se à extracção de um vector, contendo os valores de pitch da melodia entre intervalos de  $w$  segundos começando no tempo de início da nota inicial. Mais precisamente os valores de pitch do vector foram retirados nos tempos dados por  $b_1 + j \frac{w}{d-1}$ , onde  $j = 0, \dots, d - 1$ .

1 e  $d$  é o comprimento do vector. Para tornar este método independente da transposição, os vectores são normalizados para apresentarem um valor médio de zero.

No entanto, esta abordagem, embora conceptualmente simples, apresenta uma probabilidade de erro associada muito elevada, como se pode observar na Figura 5. Um simples desvio na sequência de notas faz com que os vectores de pitch não sejam coincidentes, provocando a existência de falsos negativos.

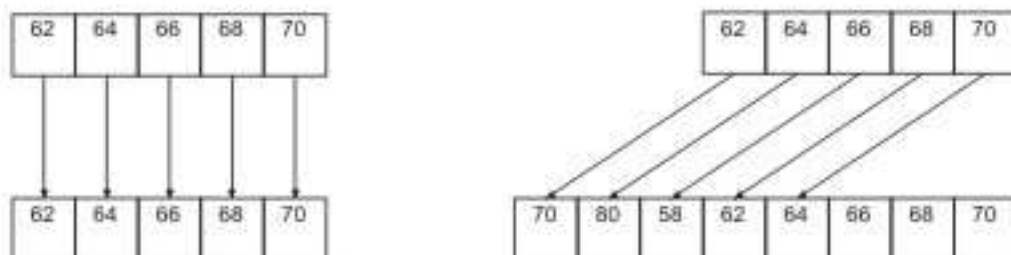


Figura 5. Vectores de Pitch de duas melodias (antes da normalização).

Para melhorar esta representação pode-se utilizar um método similar ao sugerido por Matti Ryynänen e Anssi Klapuri em [4]. Em vez de representar a melodia na sua totalidade, esta é representada por vários vectores de pitch, denominados fragmentos melódicos, como se pode observar na figura 6 (retirada de [4]). Para mais detalhes sobre os fragmentos melódicos, consultar [4].

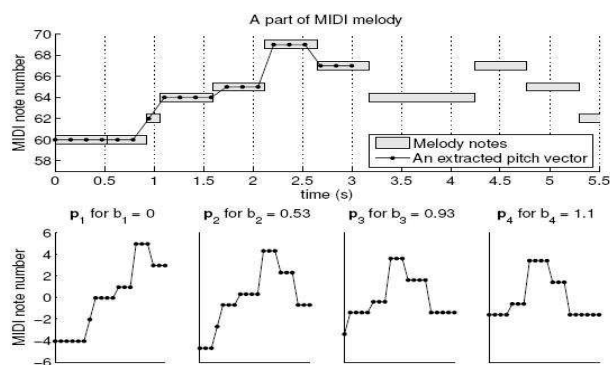


Figura 6. Extração de vectores de pitch.

### 3.1.3 Conjuntos de pontos ponderados de notas melódicas

A forma de representar a música como um conjunto de pontos ponderados é uma técnica usada com alguns séculos [9]. Nesta representação, cada nota (pontos) está inserida num espaço euclidiano, e cada ponto tem um peso associado. As dimensões são o pitch e o tempo de início da nota (“onset time”).

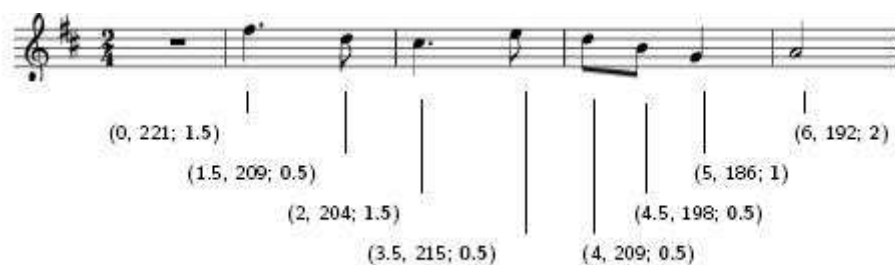


Figura 7. Exemplo de uma música representada com o conjunto de pontos ponderados.

Em [5][10] a melodia é então representada como um conjunto de triplos  $(t, p, w)$ , onde  $t$  é o tempo,  $p$  o pitch e o  $w$  o peso. Na Figura 7, retirada de [5], é ilustrada esta representação onde o peso reflecte a duração da nota.

### 3.2 Timbre

Timbre é a característica de uma nota musical, tom ou som, que permite a percepção aos ouvintes da diferença de duas notas com o mesmo pitch e intensidade de som. O timbre depende da percepção da qualidade do som, a qual está relacionada com os instrumentos, possíveis efeitos e técnicas sonoras usadas. Para melhor compreender esta dimensão, consideremos o seguinte exemplo: uma flauta e um oboé tocam a mesma nota, com a mesma duração e mesma intensidade, no entanto é possível efectuar a distinção dos sons. Isto é causado pelo facto que cada nota de cada instrumento ser um sinal complexo contendo várias frequências [26].

A percepção do timbre relaciona-se com a estrutura do espectro de um sinal, ou seja, pela sua representação no domínio das frequências. Sendo a evolução temporal do som também uma característica importante, uma forma comum de representação do timbre é feita num espectrograma, com o tempo no eixo das abcissas, a frequência no eixo das ordenadas e a cor de cada elemento representa a energia (Figura 8, retirada de domínio publico).

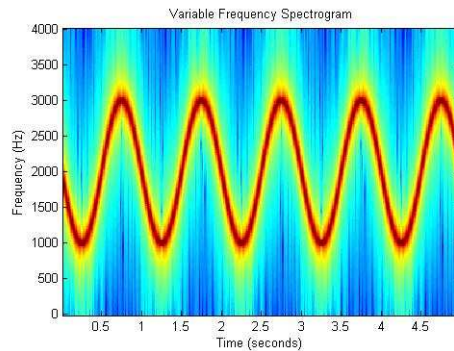


Figura 8. Exemplo de um espectrograma.

No entanto esta representação não é adequada para ser usada como um descritor devido ao seu número elevado de dimensões [15]. Em [15][22] é mencionado que diversos estudos apontam os parâmetros “attack time”, “spectral centroid” e “spectral flux” como descritores relevantes do timbre. Um outro descritor bastante utilizado nas diferentes aplicações na área MIR, é o conjunto dos coeficientes da frequência de MEL (MFCC – Mel-Frequency Cepstral Coefficients). Estes coeficientes foram primeiramente utilizados na área de reconhecimento de discurso, e em 1997 [16] foram introduzidos no campo MIR. Posteriormente, Beth Logan [17] realizou um estudo em que demonstrou a aplicabilidade dos MFCCs na representação de música. Como uma grande parte dos projectos presentes na área de recuperação de informação musical (MIR) se baseiam nestes coeficientes, será explicado brevemente o processo de obtenção dos MFCCs. É importante também de referir que um estudo realizado em [39] refere que os MFCCs apresentam um melhor resultado que os descritores presentes no MPEG-7 para diversas tarefas na área de MIR.

### 3.2.1 Mel-Frequency Cepstral Coefficients – MFCCs

Os “Mel-Frequency Cepstral Coefficients”, embora reflectam apenas a dimensão timbre, desempenham um papel importante no campo da análise de semelhança de música, uma vez que se acredita que os ouvintes são bastante sensíveis ao timbre de uma música [15]. Como foi referido na secção anterior, o timbre está relacionado com o espectro da amplitude do som/sinal, sendo este um factor de sucesso do uso de MFCCs, na medida em que estes permitem uma representação compacta do espectro. Na Figura 9 é ilustrado o processo de cálculo dos MFCCs.

O cálculo dos MFCCs a partir do sinal/som inclui as seguintes fases:



1. Divisão em várias frames do sinal de áudio, normalmente com 20-25 milissegundos de duração.
2. Aplicação de uma “window function” a cada frame para remover efeitos colaterais associados à divisão do áudio em frames.
3. Utilização de uma transformada discreta de Fourier (DFT) a cada frame de áudio do domínio de tempo para domínio de frequência. Deste modo o espectro da amplitude  $P(f)$  é obtido.
4. Introdução da transformação para “Mel-Frequency”, designada por “Mel-Scaling”.

$$M(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right)$$

5. Remoção do logaritmo.
6. Emprego da transformada discreta de co-senos (DCT) ao espectro das “Mel-Frequency” para obter um vector com os MFCCs.



Figura 9. Diagrama do cálculo de MFCCs.

Para um maior detalhe sobre este processo de extracção dos MFCCs consultar [27][61].

Como exemplos de trabalhos de investigação na área de MIR que usam os MFCCs surgem, entre os mais mencionados, os projectos BL[18], AP[19], Pam[20], GT[22] e ME[21].

### 3.3 Ritmo

De uma forma geral, o ritmo designa o tempo que demora a suceder-se um qualquer acontecimento repetitivo, sendo normalmente empregue na área da música, da dança e parte da poesia.

Na área de música, ritmo é um acontecimento sonoro relacionado com a repetição periódica, com possíveis pequenas variações, de padrões temporais de sons. Estes sons não necessitam de ter um pitch reconhecível, uma vez que o reconhecimento

ou percepção de ritmo baseia-se no início de notas musicais ou de outros sons (“onset time”).

É normalmente assumido que o ritmo representa a dimensão mais fácil de reconhecimento entre a maioria dos ouvintes. Esta suposição é baseada na música ocidental, onde existe uma menor evolução rítmica na linguagem musical. Tomando a música africana ou da Europa de leste como exemplo, a informação rítmica apresenta uma maior complexidade, onde várias “time signatures” são usadas num mesmo trabalho musical, tornando o ritmo numa característica de várias dimensões. Este tipo de música é designado por poli ritmo[15].

No caso da música ocidental, o ritmo é baseado em variantes de uma estrutura com um número fixo de “beats” igualmente espaçadas. Mais especificamente, considerando o género rock, constata-se que esta estrutura possui quatro “beats” onde a primeira e a terceira são mais fortes que a segunda e a quarta [15]. Deste modo, torna-se relevante a velocidade com que as “beats” são realizadas. Tendo como objectivo a percepção da estrutura rítmica, várias aproximações para a sua extracção foram desenvolvidas no campo da MIR, as quais se baseiam no cálculo de uma matriz de auto-semelhança [15] ou de histogramas de “beat” [22], de modo a obter a estrutura periódica da música.

Em 2001, Jonathan Foote desenvolveu uma estrutura baseada numa matriz de auto-semelhança para descrever a periodicidade rítmica de música, a qual é designada por “beat spectrum” [25]. Esta estrutura tem sido utilizada em vários trabalhos que abordam a semelhança rítmica [23][24] com resultados satisfatórios. Na secção seguinte será explicado de forma simplificada o processo de criação da estrutura “beat spectrum”.

### **3.3.1 Beat Spectrum**

Em [25] é introduzido um método, “beat spectrum”, para caracterização do ritmo e do tempo de uma música. O “beat spectrum” consiste numa medida de auto-semelhança acústica em função do tempo. Música bastante estruturada ou com padrões repetitivos terá fortes picos no “beat spectrum” nos instantes de repetição. Na Figura 10, retirada de [25], pretende-se ilustrar o “beat spectrum” do trabalho musical “Gould Prelude” onde é visível a propriedade anteriormente descrita.

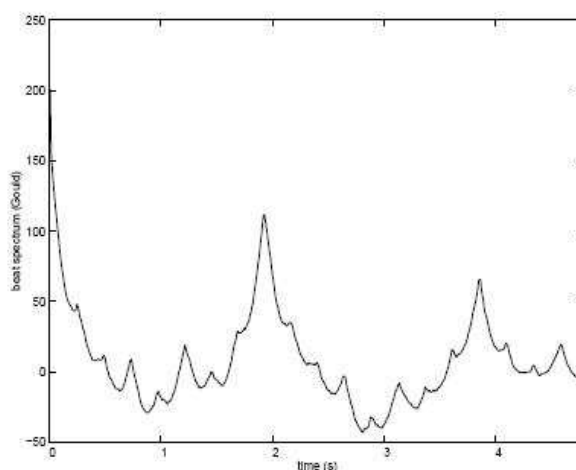


Figura 10. “Beat spectrum” de “Gould Prelude”.

O “beat spectrum” é derivado do sinal de áudio em três passos:

1. Parametrização: Neste passo o ficheiro de música é transformado numa representação espectral do sinal. A música é dividida em frames de alguns milissegundos (10 – 20) aos quais é aplicada uma transformação de Fourier e retirado o logaritmo da magnitude. O resultado consiste num conjunto de vectores de parâmetros. Em [23] foi utilizado os MFCCs para este fim.
2. Auto-semelhança: Para cada par de vectores é calculada a distância de semelhança. Em [25] é referido a distância euclidiana ou a “cousine distance”. Estas distâncias são organizadas numa estrutura  $S$  de duas dimensões (cada dimensão corresponde ao número de frames), designada de matriz de auto-semelhança, representando a estrutura do ficheiro de música. A Figura 11, retirada de [24], ilustra o processo de construção da matriz. A Figura 12, retirada de [24], esquematiza a matriz de semelhança de uma música em escala de cinzento proporcional ao valor de semelhança de cada frame. O valor de máxima semelhança é representado com o tom mais claro.

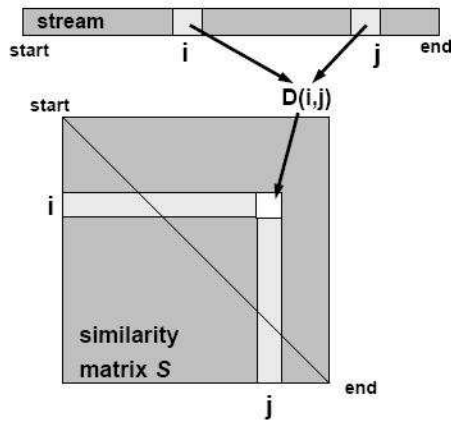


Figura 11. Matriz de auto-similaridade.

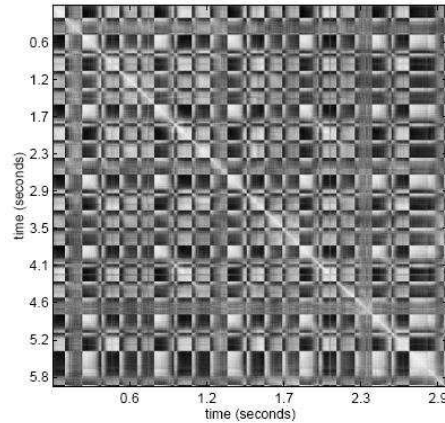


Figura 12. Matriz de auto-similaridade em tons de cinza.

3. Somatório: O “beat spectrum” resulta das periodicidades encontradas na matriz de auto-similaridade através das somas das diagonais. Dada a matriz de auto-similaridade  $S$  e dado um intervalo de tempo  $l$ , o “beat spectrum”  $B(l)$  é dado por:

$$B(l) = \sum_{k \in R} S(k, k + l)$$

onde  $R$  é um intervalo contínuo sobre as frames.

O resultado deste processo é um vector que modela os padrões rítmicos de uma música em função do tempo.

### 3.4 Conclusões e contributos

Neste capítulo foram descritas as três dimensões mais importantes da música (melodia, timbre e ritmo), assim como as diferentes técnicas para a sua representação. Relativamente às dimensões é possível concluir que:

- A dimensão melodia é mais utilizada para aplicações de consulta de músicas semelhantes a uma dada voz [3]. Normalmente é utilizada música no formato simbólico Midi monofónico, uma vez que simplifica a transcrição da melodia do que nos formatos Midi polifónico e áudio.
- A dimensão timbre é mais utilizada para aplicações de similaridade de música e geração de “playlists” [19][22][23][40][61]. Nesta dimensão é utilizada música no formato de áudio. No entanto, constata-se que o seu processamento é computacionalmente pesado, quer a nível temporal ou a

nível espacial. Os MFCCs constituem a técnica mais utilizada para descrever o timbre.

- Na dimensão ritmo não foi encontrado um número significativo de trabalhos de investigação. No entanto, com base em [22], [23] e [24] pode-se afirmar que é usado o formato de áudio e que à semelhança do timbre, o seu processamento temporal e espacial é computacionalmente pesado.

Relativamente ao estudo das técnicas para a representação de música, é importante de referir que inicialmente foi colocada a hipótese de utilizar as três dimensões da música. No entanto, foram utilizadas apenas as dimensões melodia e timbre, uma vez que existe uma maior investigação na área de MIR utilizando estas dimensões.



## 4. Semelhança de músicas

Para poder falar de músicas semelhantes é necessário definir critérios de semelhança entre músicas. Estes critérios de semelhança serão usados tendo como base as dimensões usadas na representação da música. De acordo com a representação da música existem diferentes funções para determinar o nível de semelhança entre duas músicas. O resultado destas funções é um número pertencente a  $\mathbb{R}_0^+$ , onde quanto mais próximo de zero, maior a semelhança.

Nas secções seguintes são descritas várias funções usadas em alguns dos trabalhos mais relevantes na área de MIR. Algumas das funções descritas caracterizam-se por serem métricas. Formalmente, sendo  $U$  o conjunto de elementos (universo), uma função  $d: U \times U \rightarrow \mathbb{R}_0^+$  é métrica se satisfizer as seguintes três propriedades:

- (1) Positiva:  $d(x, y) \geq 0$  e  $d(x, y) = 0 \Leftrightarrow x = y$ ;
- (2) Simétrica:  $d(x, y) = d(y, x)$ ;
- (3) Desigualdade triangular:  $d(x, y) \leq d(x, z) + d(z, y)$ .

### 4.1 Distância de edição

A função distância de edição mede a distância entre duas strings, sendo o seu valor igual ao número de operações necessárias para transformar uma string na outra, com o respectivo custo. As operações consideradas são a inserção, eliminação e substituição de elementos.

**Definição de distância de edição:** Dadas duas strings  $R$  e  $S$  de comprimento  $M$  e  $N$ , respectivamente, a distância entre ambas, denotada por  $ED(R, S)$  é definida como:

$$ED(R, S) = \begin{cases} M \times gap & \text{se } N = 0; \\ N \times gap & \text{se } M = 0; \\ \min \left\{ \begin{array}{l} ED(Rest(R), Rest(S)) + cost(first(R), first(S)), \\ ED(Rest(R), S) + gap, \\ ED(R, Rest(S)) + gap \end{array} \right\} & \text{default} \end{cases}$$

onde *gap* representa o custo de adicionar ou eliminar elementos nas strings e *cost* a função que mede a distância entre substituir dois elementos. Os operadores *first* e *Rest* representam o primeiro carácter da string e todos os outros restantes, respectivamente.

Com base em [1], se a função custo (*cost*) associado à substituição é métrica, então a função distância de edição também o é.

Em [1], a melodia é representada como uma sequência de intervalos de notas (strings) e é utilizada a distância de edição para medir a semelhança entre as strings melódicas de duas músicas. Neste caso, o *gap* tem valor de 1 e o custo associado à substituição é dado pela função de distância entre intervalos de notas. Sejam  $I_R = (x_{lir}, x_{pitch})$  e  $I_S = (y_{lir}, y_{pitch})$  dois intervalos de notas de duas strings melódicas respectivamente, então a distância entre intervalos  $d(I_R, I_S)$  é definida da seguinte forma:

$$d(I_R, I_S) = a|x_{lir} - y_{lir}| + b|x_{pitch} - y_{pitch}|$$

É provado em [1] que esta função é métrica. Os valores não negativos *a* e *b* foram determinados com o objectivo de otimizar a performance, tendo como base as músicas usadas.

Assim sendo, a função de semelhança entre as melodias  $R = \langle I_{R1}, \dots, I_{Rn} \rangle$  e  $S = \langle I_{S1}, \dots, I_{Sm} \rangle$  é dada por:

$$MED(R, S) = \begin{cases} M & \text{se } N = 0; \\ N & \text{se } M = 0; \\ \min \left\{ \begin{array}{l} MED(Rest(R), Rest(S)) + d(first(R), first(S)), \\ MED(Rest(R), S) + 1, \\ MED(R, Rest(S)) + 1 \end{array} \right\} & \text{caso contrário} \end{cases}$$



## 4.2 L<sub>p</sub> Norm

Esta função é normalmente usada em sequências de valores com a mesma dimensão.

**Definição de norma de distância L<sub>p</sub>:** Dadas duas séries de valores  $R = \langle r_1, \dots, r_d \rangle$  e  $S = \langle s_1, \dots, s_d \rangle$  com comprimento  $d$ , a distância entre ambas é definida por:

$$L_p \text{Norm}(R, S) = \sqrt[p]{\sum_{i=1}^d (r_i - s_i)^p}$$

Quando  $p = 1$  esta distância é chamada por distância de Manhattan ou “city-block”, e quando  $p = 2$  é designada como distância euclidiana. Todas as funções do conjunto da L<sub>p</sub> Norm são métricas [13, secção 2.2.1].

Em [4] a melodia de uma música é descrita por uma sequência de pitch de tamanho  $d$ . O nível de semelhança entre melodias é calculado usando a distância euclidiana. Caso este valor seja inferior a um determinado limite podemos afirmar que as duas melodias são semelhantes, ou seja, as músicas apresentam um grau aceitável de parecença. Sejam  $R$  e  $S$  duas sequências de pitch de tamanho  $d$ , então:

$$\text{dist}(R, S) = \sqrt[2]{\sum_{i=1}^d (r_i - s_i)^2}$$

Em [61], Alfie Tan Kok Leong propõe um método para o cálculo de semelhança de músicas, em que a sua caracterização é feita através de uma assinatura calculada a partir dos MFCCs, a qual é posteriormente comparada com outras assinaturas pela distância L2, vulgarmente designada, distância euclidiana.

Cada música é dividida em frames de 25,6 milissegundos, onde cada segmento sobrepõe o anterior em 15.6 milissegundos. Para cada frame é obtido um vector com os coeficientes de “Mel-Frequency”(MFCCs) (são usados 13 coeficientes) obtendo deste modo uma descrição da música com um determinado número de frames de MFCCs, os quais descrevem 25,6 milissegundos do espectro do áudio. Devido ao elevado número de descritores, os parâmetros MFCC são sujeitos a uma redução no tamanho dos dados, tendo como objectivo uma maior eficiência de computação e um menor espaço de armazenamento. A redução dos dados é efectuada através do agrupamento dos valores

de diferentes frames, de modo a obter uma assinatura de menor dimensão de uma música. O algoritmo de agrupamento, ou do termo em inglês “clustering”, utilizado é o algoritmo K-Means [61 secção 8].

Em [24] o ritmo de uma música é representado pelo seu “beat spectrum”. Dadas duas músicas  $M_1$  e  $M_2$  e os seus respectivos “beat spectrum”  $B_1$  e  $B_2$  de tamanho  $L$ , o nível de semelhança entre músicas é calculado usando a distância euclidiana entre os dois vectores de “beat spectrum”. Caso este valor seja inferior a um determinado limite podemos afirmar que as duas músicas são semelhantes.

$$dist(B_1, B_2) = \sqrt{\sum_{i=1}^L (B_{1i} - B_{2i})^2}$$

### 4.3 Distância de transportação

O termo método de transportação é utilizado, na área da matemática, para o estudo da forma óptima de alocação de recursos [28][29]. O problema foi introduzido pelo matemático francês Gaspard Monge em 1771 e ao longo do tempo sofreu avanços por outros investigadores, entre os quais Kantorovich se destacou recebendo o premio Nobel da economia por aplicação do problema de transportação [28].

Em [9], é referido que a distância de transportação tem as seguintes vantagens:

- Contínua: Se as diferenças entre dois conjuntos são pequenas, a distância de transportação será um valor relativamente pequeno. Quando um conjunto sofre uma pequena distorção a distância não irá sofrer uma grande alteração.
- Suporte para distorções: Vários tipos de diferenças, como por exemplo neste caso, as diferenças de pitch, duração de notas e tempo são tidas em conta sem a necessidade de as antecipar explicitamente, ou seja, não é necessário nenhuma alteração da função, uma vez que esta já apresenta suporte para distorções.
- Unificação parcial: Para algumas distâncias de transportação, caso um dos conjuntos tenha menos pontos que o outro, a distância de transportação automaticamente selecciona os pontos que melhor encaixam no outro conjunto, ignorando os pontos do conjunto maior que não têm correspondentes no conjunto menor.

- Flexibilidade: As distâncias de transporte permitem ajustes e afinações oferecendo a possibilidade de alteração do esquema de pesos e da distância base.

Nesta secção são descritas duas funções de transporte usadas em trabalhos relevantes na área de MIR: a “Earth Mover Distance” e a “Proportional transportation distance”.

#### 4.3.1 Earth Mover Distance

A “earth mover’s distance”(EMD)[6] entre dois conjuntos de pontos ponderados mede o esforço necessário para transformar o primeiro conjunto no segundo. Intuitivamente, dadas duas distribuições fictícias: dado um determinado espaço, o primeiro conjunto poderia ser uma quantidade  $n$  de terra dispersa e o segundo conjunto a quantidade de buracos presentes. A EMD mede a quantidade de trabalho necessária para mover terra de modo a encher os buracos, onde cada unidade de trabalho corresponde a transportar uma unidade de terra por cada unidade de distância.

**Definição de “earth mover’s distance”:** Seja  $A = \{a_1, \dots, a_m\}$  um conjunto de pontos ponderados tal que  $a_i = \{(x_i, w_i)\}$ ,  $i = 1, \dots, m$ , onde  $x_i \in \mathbb{R}^k$  e  $w_i \in \mathbb{R}$  correspondendo ao respectivo peso. Seja  $W = \sum_{j=1}^m w_i$  o peso total do conjunto  $A$ . Dados dois conjuntos de pontos ponderados  $A$  e  $B$ , uma distância base  $d$  em  $\mathbb{R}^k$  e  $f_{ij}$  o fluxo do peso de  $x_i$  para  $y_j$  através da distância  $d_{ij}$ . Se  $W, U$  são os pesos totais de  $A$  e  $B$  respectivamente, o conjunto  $\mathcal{F}$  de todos os possíveis fluxos  $F = [f_{ij}]$  é definido pelas seguintes restrições:

1.  $f_{ij} \geq 0, i = 1, \dots, m, j = 1, \dots, n$   
Cada fluxo é um número não negativo.
2.  $\sum_{j=1}^n f_{ij} \leq w_i, i = 1, \dots, m$   
Nenhum ponto transfere mais peso do que possui.
3.  $\sum_{i=1}^m f_{ij} \leq u_j, j = 1, \dots, n$   
Nenhum ponto recebe mais peso do que precisa.
4.  $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(W, U)$   
O peso total transportado é o mínimo dos pesos totais dos dois conjuntos.

O fluxo do peso  $f_{ij}$  sobre a distância  $d_{ij}$  é caracterizado pelo produto deste com a distância. A soma de todos estes produtos é o custo total de transformar  $A$  em  $B$ . A

$EMD(A, B)$  é definida sobre o custo total mínimo sobre  $\mathcal{F}$ , normalizado pelo peso do conjunto com menor peso, da seguinte forma:

$$EMD(A, B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\min(W, U)}$$

Esta função tem as seguintes propriedades (a sua prova pode ser encontrada em [6]):

1. A EMD é métrica se  $d$  é métrica e se os pesos totais são iguais;
2. É contínua, ou seja, uma pequena mudança nos valores de dimensão ou peso, reflecte apenas uma pequena alteração no valor final. Esta propriedade é importante caso existam distorções aquando da transcrição da melodia da música;
3. Não obedece às propriedades de positividade e desigualdade triangular se os pesos totais não forem iguais.

A EMD pode ser computada eficientemente como um problema de programação linear usando uma versão do algoritmo SIMPLEX[7].

#### 4.3.2 Proportional transportation distance

Face à restrição existente na propriedade da desigualdade triangular na EMD e a vantagem que esta propriedade apresenta em acelerar o processo de pesquisa, (como será explicado no capítulo 5). Giannopoulos e Veltkamp propõem em [8] uma modificação à EMD de modo a considerar os conjuntos de pontos ponderados com pesos totais diferentes, mantendo a propriedade da desigualdade triangular. Esta nova função designa-se por “proportional transportation distance”(PTD), e tem como objectivo remover a diferença do peso total, de modo a manter as proporções dos pesos antes da aplicação da função EMD. O primeiro passo consiste em dividir, para os dois conjuntos de pontos, o peso de cada ponto pelo peso total do respectivo conjunto e de seguida aplicar a função EMD para os conjuntos resultantes.

**Definição de “proportional transportation distance”:** Considerando  $A$  e  $B$  dois conjuntos de pontos ponderados,  $W$  e  $U$ , os respectivos pesos totais, e  $d$  uma distância base, o conjunto  $\mathcal{F}$  de todos os fluxos possíveis  $F = [f_{ij}]$  de  $A$  para  $B$  é definido pelas seguintes restrições:

1.  $f_{ij} \geq 0, i = 1, \dots, m, j = 1, \dots, n$   
Cada fluxo é um número não negativo.

2.  $\sum_{j=1}^n f_{ij} = w_i, i = 1, \dots, m$
3.  $\sum_{i=1}^m f_{ij} = \frac{u_j W}{U}, j = 1, \dots, n$
4.  $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = W$

O fluxo do peso  $f_{ij}$  sobre a distância  $d_{ij}$  é caracterizado pelo produto deste com a distância. A soma de todos estes produtos é o custo total de transformar  $A$  em  $B$ . A  $PTD(A, B)$  é definida sobre o custo total mínimo sobre  $\mathcal{F}$ , normalizado pelo peso do conjunto  $W$ , da seguinte forma:

$$PTD(A, B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{W}$$

As restrições 2 e 4 garantem que todo o peso de  $A$  é movido para os pontos em  $B$  e a restrição 3 garante que isto é realizado respeitando as antigas percentagens do peso em  $B$ . Para um maior nível de detalhe sobre esta distância de transportação consultar [8].

Em [5], a transcrição da melodia em conjuntos de pontos ponderados permite a aplicação de “transportation distance” para o cálculo de semelhança. Neste trabalho são usadas a “earth mover’s distance”(EMD) e a “proportional transportation distance”(PTD).

Neste método, por uma questão de simplicidade, ambas as funções usam a distância euclidiana como distância de base - Sendo assim a distância entre duas notas com coordenadas  $(t_1, p_1)$  e  $(t_2, p_2)$  é dada como:

$$\sqrt{(t_1 - t_2)^2 + (p_1 - p_2)^2}$$

Antes da aplicação do cálculo da distância de transportação, em [5] foram realizados os seguintes ajustes ao conjunto de pontos ponderados:

1. Para ser possível identificar a semelhança entre versões aumentadas ou diminuídas de melodias (diferente tempo da última nota), é necessário normalizar as coordenadas de tempo. Assim sendo é escolhida a melodia com o menor tempo da última nota para esticar, ou seja, multiplicar as coordenadas do tempo por um certo factor, para que ambas as melodias possuam o mesmo valor de coordenada de tempo final. Este alinhamento é realizado aquando do uso da PTD. No caso da EMD, é calculada a semelhança com e sem este ajuste, usando-se o valor mínimo obtido.

2. Para tornar a medida independente das transposições, é realizado uma transposição duma melodia na escala de pitch para uma posição onde a distância é mínima. No entanto, obter esta transposição mínima é um cálculo muito custoso, pelo que, o processo usado consistiu em obter uma transposição duma melodia em que a média do pitch é igual. No entanto, é dito no artigo que isto não garante uma transposição mínima.
3. No cálculo da distância de transportação, a substituição do peso de uma nota para a outra pode acontecer na dimensão tempo, pitch ou em ambas. Em [5], foi realizado, na dimensão do tempo a multiplicação das coordenadas por um valor 3. Este valor foi obtido experimentalmente por testes e afinação do algoritmo.

A Figura 13 (retirada de [9]) mostra um exemplo de duas melodias após os ajustes acima mencionados nos conjuntos de pontos ponderados. Nesta figura, é de mencionar que sem os ajustes, as duas melodias possuem uma distância de 5.41825, enquanto que com os ajustes apenas distam 0.739529 [5].

Nesta figura, cada seta indica o fluxo e o peso transportado entre cada par de notas.

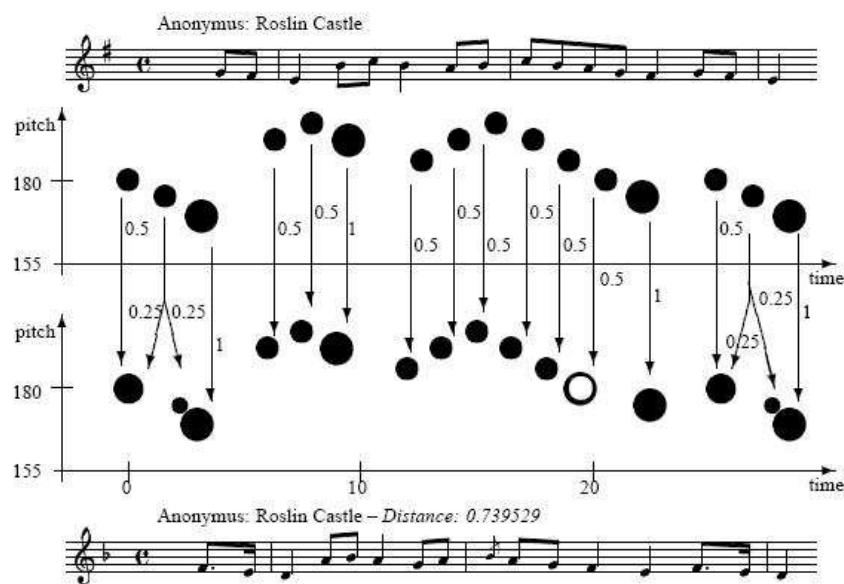


Figura 13. Ilustração do fluxo de pesos usando a EMD.

Rainer Typke em [9] estudou várias melhorias a este método, com as quais concorreu nas conferências MIREX 2005 e MIREX 2006, obtendo resultados bastante satisfatórios. Para mais detalhes sobre as inovações propostas e resultados obtidos consultar [5][6][7].

Em [18], Beth Logan e Ariel Salomon calculam a semelhança de músicas baseada no espectro do áudio. Por terem sido os pioneiros, o método é referido por LS na área de investigação MIR. Este método analisa a informação presente no espectro que corresponde à dimensão timbre da música.

Cada música é representada por uma assinatura baseada nos seus MFCCs e o cálculo da medida de semelhança entre duas assinaturas é efectuada com base na distância de transportação, “earth mover’s distance”(EMD)[2].

Na Figura 14 é ilustrado o processo de cálculo da assinatura. Cada música é dividida em frames de 25,6 milissegundos em que cada um sobrepõe o anterior em 10 milissegundos. Para cada frame é obtido um vector com os coeficientes de “Mel-Frequency”(MFCCs) (apenas 13 a 30 coeficientes são usados), obtendo deste modo uma descrição da música com um número constante de frames de MFCCs, os quais descrevem 25,6 milissegundos do espectro do áudio. Por último, é obtida uma assinatura mais compacta da música através de classificação das frames obtidas na música em 16 agrupamentos. As frames da música que forem mais semelhantes, de acordo com a distância euclidiana, são agrupadas no mesmo agrupamento. Após este processo, cada música é representada por um conjunto de 16 triplos  $\langle \mu, \Sigma, w \rangle$ , sendo cada triplo associado a um dado agrupamento. Onde  $\mu$  é o valor médio,  $\Sigma$  a co-variância e  $w$  o peso, que corresponde ao número de frames contidas no agrupamento.



Figura 14. Diagrama do cálculo da assinatura de uma música.

A semelhança entre duas assinaturas de música é então medida pela EMD. Dadas duas assinaturas  $P = \{(\mu_{p1}, \Sigma_{p1}, w_{p1}), \dots, (\mu_{p16}, \Sigma_{p16}, w_{p16})\}$  e  $Q = \{(\mu_{q1}, \Sigma_{q1}, w_{q1}), \dots, (\mu_{q16}, \Sigma_{q16}, w_{q16})\}$  onde  $\mu$  corresponde ao valor médio,  $\Sigma$  corresponde à co-variância e  $w$  ao peso de cada agrupamento, a distância base  $d_{p_i q_i}$  corresponde à versão simétrica da distância “Kullback-Leibler (KL)”:

$$d_{p_i q_j} = \frac{\sum p_i}{\sum q_j} + \frac{\sum q_j}{\sum p_i} + (\mu_{p_i} - \mu_{q_j})^2 \times \left( \frac{1}{\sum p_i} + \frac{1}{\sum q_j} \right)$$

#### 4.4 Dynamic time warping (DTW)

O função “dynamic time warping”[12] permite a medição de semelhança entre duas séries temporais de tamanhos diferentes, encontrando o melhor alinhamento entre ambas.

**Definição “Dynamic time warping”:** Dadas duas séries temporais  $R = \langle r_1, \dots, r_M \rangle$  e  $S = \langle s_1, \dots, s_N \rangle$  de comprimento  $M$  e  $N$ , respectivamente, a distância DTW é definida por:

$$DTW(R, S) = \begin{cases} \infty & \text{se } M = 0 \text{ ou } N = 0 \\ dist(r_1, s_1) + \min \begin{cases} DTW(Rest(R), Rest(S)), \\ DTW(Rest(R), S), \\ DTW(R, Rest(S)) \end{cases} & \text{default} \end{cases}$$

onde  $dist(r_1, s_1) = |r_1 - s_1|$  e  $Rest(R)$  representa a subsequência  $R$  sem o primeiro elemento,  $\langle (t_2, r_2), \dots, (t_M, r_M) \rangle$ .

A função de distância pode ser uma das  $L_p$ -Norms, como por exemplo a distância euclidiana ( $p = 2$ ). Apesar de usar uma função métrica como medida entre dois elementos, a DTW não obedece à desigualdade triangular (consultar [13] para observar a demonstração).

Em [14], a melodia de uma música é transcrita para uma sequência de pitch. De modo a obter esta sequência, a melodia é dividida em frames de 100 milissegundos. Podendo existir várias notas na mesma frame, neste caso o pitch extraído é aquele que tenha um valor de pitch mais contido no intervalo. De forma a tomar em conta a variação do tempo nas músicas, este processo é repetido com diferentes factores de escala.

Após a transcrição obtém-se uma sequência de valores que representa a música. Uma vez que normalmente as músicas possuem durações diferentes, isto implica que as séries resultantes possuam comprimentos diferentes. Face a esta restrição, é utilizada a função “dynamic time warping” para obter o valor de semelhança entre as músicas.

#### 4.5 Conclusões e contributos

Neste capítulo foram descritas várias medidas de semelhança utilizadas em diversos trabalhos de semelhança de música, utilizando as técnicas apresentadas no capítulo 3 para descrever o conteúdo musical.

Das medidas apresentadas, existem algumas que são métricas, sendo estas as consideradas para a possível constituição de espaços métricos. As medidas que



cumprem os requisitos de métrica são: a distância de edição, as medidas da família  $L_p$ Norm e a “earth mover’s distance”. Esta última possui diversas restrições para que constitua uma verdadeira métrica. Com base nestas métricas e nas técnicas apresentadas no capítulo 3, conclui-se que é possível definir espaços métricos.

Por outro lado, conclui-se ainda que a escolha de determinada medida é condicionada pela representação utilizada para o conteúdo musical. As métricas da  $L_p$ Norm são apenas aplicáveis em sequências numéricas de tamanho fixo e mesma dimensão. Por outro lado, a “earth mover’s distance” é aplicável na semelhança entre conjuntos.

Relativamente à constituição de espaços métricos, nesta dissertação foram seleccionados três espaços. Um na dimensão melodia, utilizando a representação de intervalos melódicos e a distância de edição. Os outros dois na dimensão timbre, utilizando uma assinatura baseada nos MFCCs e as distâncias de Manhattan e euclidiana.



## 5. Pesquisa aproximada em espaços métricos

Realizar pesquisas com base na proximidade dos elementos, pode levar-nos a uma pesquisa exaustiva, uma vez que é necessário calcular a distância entre o elemento de pesquisa e cada elemento da base de dados. Para além da pesquisa exaustiva, existe a condicionante de que em alguns casos o cálculo da distância é muito custoso, o que diminui a eficácia da pesquisa. Tendo em conta as diferentes formas de representação e funções para o processamento da música anteriormente mencionados, é perceptível que quanto maior o número de dimensões da música usado melhor o resultado obtido, mas maior o custo por cada cálculo de distância.

Sendo o nosso objectivo a pesquisa aproximada em espaços métricos, é de notar que existem várias estruturas métricas de dados que tornam a pesquisa mais eficiente. Estas estruturas de dados são indexadas por distância e permitem particionar o conjunto de elementos em subconjuntos disjuntos, tais que os elementos no mesmo subconjunto são próximos localmente. Esta partição do espaço é feita com o objectivo de, no momento da pesquisa, só algumas zonas (subconjuntos) serem pesquisadas. Permitem também reduzir o número de cálculos de distância entre o elemento de pesquisa e os elementos pertencentes a todo o espaço. No entanto para a aplicação de tais estruturas é necessário que a função usada para medir a similaridade seja métrica, uma vez que os métodos empregues utilizam as propriedades da função para reduzir o número de cálculos de distância.

### **Espaço métrico**

Um espaço métrico é um par  $(U, d)$ , onde  $U$  é o conjunto de elementos, denominado universo, e  $d: U \times U \Rightarrow \mathbb{R}_0^+$  é uma função, denominada distância, que é métrica (secção 4).

### **Classificação de pesquisas**

As pesquisas aproximadas podem ser classificadas em uma de duas classes, pesquisa por alcance (“range query”), ou pesquisa dos  $k$  vizinhos mais próximos (“ $k$  nearest neighbor query”).

Na pesquisa por alcance o objectivo consiste em, dado um elemento de pesquisa  $q$  e um valor real não negativo  $r$  (normalmente designado por raio da pesquisa), encontrar o conjunto de elementos cuja distância ao elemento de pesquisa seja inferior ou igual ao raio de pesquisa. Formalmente, sendo  $U$  o conjunto de elementos da base de dados, temos:  $(q, r) = \{u \in U : d(q, u) \leq r\}$ .

Na pesquisa dos  $k$  vizinhos mais próximos, o objectivo consiste, dado um valor inteiro positivo  $k$ , encontrar um conjunto de tamanho  $k$ , de elementos que estejam mais próximos do elemento de pesquisa  $q$ . Formalmente, sendo  $U$  o conjunto de elementos da base de dados, consiste em obter um conjunto  $A \subseteq U$  que  $|A| = k$  e  $\forall u \in A, v \in U - A, d(q, u) \leq d(q, v)$ .

Com base no objectivo desta preparação de dissertação, apenas estamos interessados na pesquisa por alcance, sendo a pesquisa dos  $k$  vizinhos mais próximos apenas mencionada aqui como curiosidade. Nas secções seguintes apenas é descrita a pesquisa por alcance para as várias estruturas de dados métricas.

### **Classificação de estruturas de dados métricas**

Uma estrutura de dados métrica particiona a base de dados de acordo com as distâncias entre um conjunto de elementos seleccionados e os restantes, com o objectivo de minimizar o número de cálculos da função de distância aquando das pesquisas. Dependendo do método utilizado para particionar a base de dados, estas estruturas de dados são classificadas como baseadas em pivôs ou em agrupamentos (termo em inglês, “clusters”)[47].

No método de partição baseado em pivôs, é seleccionado um subconjunto de elementos da base de dados, denominados pivôs, e os restantes elementos são classificados de acordo com as suas distâncias a estes. Dois elementos  $x$  e  $y$  estão na mesma zona se, para todos os pivôs definidos, a distância de  $x$  ao pivô é igual à distância de  $y$  ao pivô. Na Figura 15, retirada de [47], o espaço foi particionado com base em 2 pivôs,  $u_8$  e  $u_{11}$ . Do ponto de vista do pivô  $u_8$ , os elementos  $u_2$  e  $u_4$  são próximos (estão à mesma distância). Assim sendo, pesquisar os elementos próximos

dum dado elemento  $q$ , é pesquisar os elementos nas intersecções de algumas esferas centradas nos pivôs. No exemplo ilustrado, os elementos candidatos ao resultado seriam  $u5$  e  $u12$ , para um raio  $r$ .

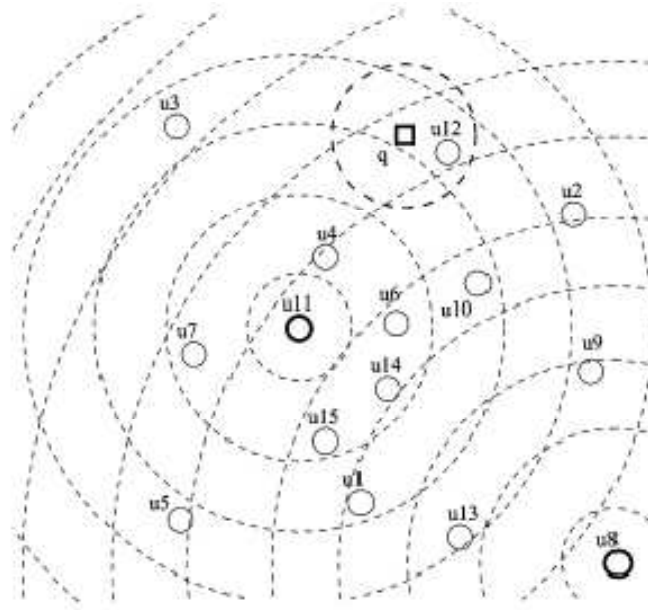


Figura 15. Partição do espaço com base em 2 pivôs  $u8$  e  $u11$ .

No método de partição baseado em agrupamentos, a base de dados é particionada em zonas, denominadas agrupamentos, que estão definidas com base na distância a um grupo de elementos, denominados centros dos agrupamentos. Ou seja, uma vez seleccionados os centros dos agrupamentos, os elementos num dado agrupamento são aqueles que estão mais próximos do seu centro do que de qualquer outro centro de agrupamento. O centro desempenha um papel muito parecido ao pivô, mas a diferença reside em que um dado elemento  $x$  está associado a um pivô  $p$  com base na distância entre  $p$  e  $x$  e não porque  $p$  seja o pivô mais próximo de  $x$ . Na Figura 16, retirada de [47], é apresentada uma partição do espaço em 4 zonas, sendo os centros de agrupamento  $c1, c2, c3$  e  $c4$ . Assim sendo, pesquisar os elementos próximos dum dado elemento  $q1$ , é pesquisar os elementos em zonas do espaço que intersectam a zona requerida. No exemplo ilustrado, seria pesquisar as zonas cujos centros são  $c1, c2$  e  $c3$  e o resultado seria os elementos  $u4$  e  $u11$ , para um raio  $r$ .

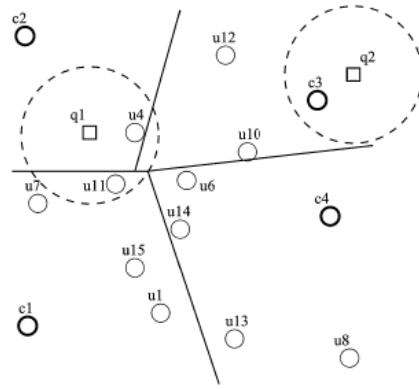


Figura 16. Partição do espaço com base em 4 agrupamentos, com centros em c1, c2, c3 e c4.

De acordo com a capacidade das estruturas de dados métricas suportarem actualizações, estas são classificadas em estruturas estáticas ou estruturas dinâmicas. Uma estrutura estática é construída no momento inicial com base num conjunto de elementos, não permitindo actualizações, i. e., inserções ou remoções. Uma estrutura dinâmica permite operações de inserção e remoção de elementos.

De seguida é efectuada uma breve descrição de algumas estruturas de dados métricas mais relevantes, já utilizadas em artigos de avaliação [57][58].

### 5.1 “List of Clusters” – LC

A estrutura de dados métrica “List of Clusters” – LC, desenvolvida por E. Chavez e G. Navarro [41], baseia-se no método de partição dos elementos em agrupamentos. Como é perceptível pelo nome, esta estrutura corresponde a uma lista de agrupamentos, onde cada agrupamento é representado por um triplo  $\langle \text{centro}, \text{raio}, \text{conteúdo} \rangle$ . O *centro* de um agrupamento corresponde a um elemento da base de dados, o *raio* é um valor real positivo pelo qual é determinado se os elementos da base de dados são inseridos no *conteúdo* do agrupamento. É importante notar que este raio não possui nenhuma relação com o raio de pesquisa anteriormente mencionado na classificação de pesquisas. Cada elemento do conjunto *conteúdo* do agrupamento possui uma distância inferior ou igual ao centro do que o raio do agrupamento. Este valor representa a similaridade existente entre o ponto central (*centro*) e o elemento. Quanto menor for esta distância maior a semelhança entre os dois elementos.

Um aspecto importante desta estrutura assenta no pressuposto de que os agrupamentos são disjuntos, ou seja, um elemento embora possua uma distância inferior

a diferentes raios de vários agrupamentos, apenas está inserido no conteúdo do primeiro agrupamento que sucede na lista. Na Figura 17 é visível esta situação, onde os elementos  $u1$  e  $u2$  estão na zona de sobreposição dos clusters  $c1$ - $c2$  e  $c2$ - $c3$ , respectivamente. Na parte direita é possível observar como a estrutura LC pode ser vista como uma lista.

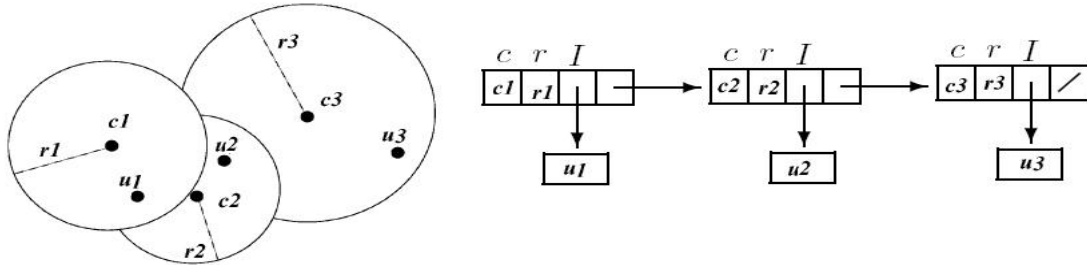


Figura 17. Representação de uma “list of clusters”.

### 5.1.1 Construção

A “list of clusters” é construída a partir de um conjunto de elementos inicial através de um processo recursivo. Dada uma base de dados de elementos  $U$  é seleccionado um determinado elemento  $c$  para centro de um agrupamento e um raio  $r_c$ . Após esta escolha é definido o conteúdo do cluster como sendo o conjunto  $I = \{u \in U - \{c\}, d(c, u) \leq r_c\}$  onde  $d(c, u)$  representa uma função de distância. Por último são retirados da base de dados os elementos inseridos do cluster, ficando esta como o conjunto  $E = \{u \in U, d(c, u) > r_c\}$  e o processo é recursivamente repetido até não existirem elementos na base de dados. Na Figura 18 é mostrada uma descrição do algoritmo.

```

LC( $U$ )
Se  $U = \{\}$ 
    Retornar lista vazia
Selecionar  $c \in U$ 
Selecionar o raio  $r_c$ 
 $I \leftarrow \{u \in U - \{c\}, d(c, u) \leq r_c\}$ 
 $E \leftarrow \{u \in U, d(c, u) > r_c\}$ 
Retornar  $(c, r, I) : LC(U)$ 

```

Figura 18. Algoritmo de construção da “list of clusters”.

O centro de um agrupamento pode ser seleccionado de acordo com as seguintes heurísticas [41]:

H1: Ao acaso;

H2: O elemento mais próximo ao centro do agrupamento anterior;

H3: O elemento mais afastado do centro do agrupamento anterior;

H4: O elemento que minimiza a soma das distâncias aos centros dos agrupamentos anteriores;

H5: O elemento que maximiza a soma das distâncias aos centros dos agrupamentos anteriores.

O raio de um agrupamento pode ser escolhido tendo em conta os seguintes aspectos [41]:

A1: Partições com raio fixo: todos os agrupamentos possuem o mesmo raio;

A2: Partições de tamanho fixo: todos os agrupamentos possuem o mesmo número de elementos e o raio é definido de acordo com a distância do elemento mais afastado do centro.

### 5.1.2 Pesquisa

A realização de uma pesquisa consiste, dado um par (elemento  $q$ , raio  $r$ ), percorrer recursivamente os agrupamentos da lista e para cada agrupamento  $(c, r_c)$  medir a distância  $d(c, q)$  e adicionar o centro ao conjunto de resposta caso a distância seja inferior ao raio de pesquisa (ver Figura 19, pesquisa  $q1$ ). Caso exista uma intersecção do espaço de pesquisa com o espaço métrico do agrupamento, é efectuada uma pesquisa exaustiva aos elementos do agrupamento (ver Figura 19, pesquisas  $q2$  e  $q4$ ). Por último, caso exista uma intersecção do espaço de pesquisa com o exterior do agrupamento, o processo é repetido recursivamente para os restantes agrupamentos da lista (ver Figura 19, pesquisas  $q3$  e  $q4$ ). De modo a solidificar estes casos, a Figura 19 mostra uma descrição do algoritmo e as representações gráficas dos casos possíveis.

```

LCpesquisa( L, q, r)
  Se  $L = \{\}$ 
    Retornar  $\{\}$ 
  Seja  $L = (c, r_c, I) : E$ 
  Se  $d(c, q) \leq r$ 
    Adicionar  $C$  à lista de resultados
  Se  $d(c, q) \leq r_c + r$ 
    Procurar exaustivamente o conj.  $I$ 
  Se  $d(c, q) > r_c - r$ 
    LCpesquisa( $E, q, r$ )

```

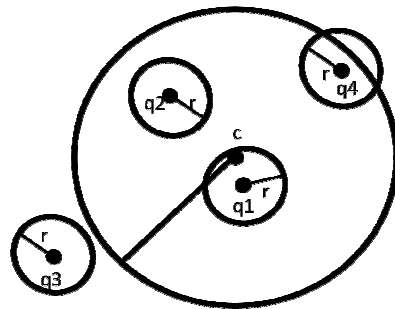


Figura 19. Algoritmo de pesquisa da “list of clusters”.



## 5.2 “Recursive List of Clusters” – RLC

A estrutura de dados métrica “Recursive List of Clusters” – RLC, desenvolvida por Margarida Mamede [42][60], baseia-se no método de partição dos elementos em agrupamentos. Esta estrutura pode ser vista como uma versão dinâmica da LC, no entanto apresenta várias particularidades na forma de organização dos elementos no interior dos agrupamentos.

Na secção anterior são referidos dois critérios para a selecção do raio dos agrupamentos. Os raios podem ter um valor fixo ou uma cardinalidade constante (possivelmente, excepto o último), no entanto o último critério apresenta a limitação de impossibilitar a execução de actualizações eficientemente. Por este motivo, a RLC apresenta um raio de comprimento fixo para os agrupamentos.

O critério de usar um valor fixo para o raio possibilita a situação de que os agrupamentos possuam um número de elementos diferente, normalmente os primeiros agrupamentos apresentam um número bastante superior de elementos, enquanto os últimos apenas possuem o centro [42]. Isto significa que, caso o raio tenha um valor reduzido para evitar esta situação, a lista de agrupamentos terá um comprimento significativo, prejudicando a eficiência das operações sobre a lista. Por outro lado, um valor grande de raio evitaria a proliferação do número de agrupamentos mas tornaria o processamento do interior dos mesmos numa operação custosa. De modo a contornar estas situações, onde ambas prejudicam a eficiência, a RLC propõe uma forma inovadora para organizar o interior de um agrupamento. A ideia consiste em implementar o interior de um agrupamento numa de duas formas: se o tamanho do interior não exceder um determinado limite (capacidade da RLC), este é implementado por um “array”. Caso contrário, este é implementado por uma “recursive list of clusters”.

Cada elemento  $x$  na base de dados possui uma sequência  $D$  com as distâncias de  $x$  aos centros dos agrupamentos aos quais  $x$  pertence. É importante mencionar que nos “arrays” são mantidas duas ordenações, os valores  $d(c, x)$  em  $D$  são ordenados de acordo com a relação pai-filho dos centros e por outro lado, cada “array” está ordenado decrescentemente pela distância de  $x$  ao centro, ou seja, pelo primeiro elemento da sequência  $D$ .

Para um valor real positivo  $r$  e um inteiro positivo  $\alpha$  (capacidade da RLC), uma “recursive list of clusters” sobre um espaço métrico pode ser vista como:  $<$

$(c_1, r, l, D_c, I_1), \dots, (c_n, r, l, D_c, I_n) >$  para um  $n \geq 0$ , e para cada  $j = 1, \dots, n$ , se o interior  $I_j$  não exceder  $\alpha$ , este é implementado por um “array”, caso contrario é implementado por uma “recursive list of clusters”. O valor  $l$  representa o nível do agrupamento que consiste no número de ancestrais do agrupamento, excluindo o próprio. Para melhor compreender esta estrutura, na Figura 20 (adaptada de [42]), é ilustrada uma representação gráfica de uma “recursive list of clusters”. Tomando como exemplo o elemento  $c_1''$ , a sua sequência  $D$  é dada por  $\langle d(c_1'', c_2'), d(c_1'', c_1) \rangle$  e para todos os elementos  $x$  presentes neste agrupamento, o “array” contém o par  $(\langle d(x, c_1''), d(x, c_2') \rangle, d(x, c_1)) >, x)$ .

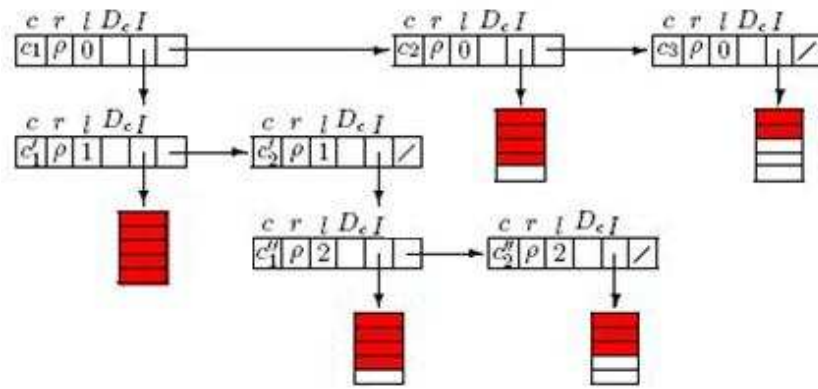


Figura 20. Representação gráfica da “recursive list of clusters”.

### 5.2.1 Construção

Sendo a RLC uma estrutura dinâmica, esta é formada por uma sequência de operações de inserção. No entanto, a sua organização depende da ordem dos elementos na inserção, ou seja, com o mesmo conjunto de elementos é possível obter diferentes RLCs se estes forem inseridos com ordem diferente.

### 5.2.2 Inserção

A inserção de um elemento é realizada percorrendo a lista, verificando se a distância entre o elemento e o centro do agrupamento é inferior ou igual ao raio do agrupamento. Caso não seja encontrado nenhum agrupamento onde o elemento seja próximo, é criado no fim da lista um novo agrupamento tendo como centro o novo elemento. No caso em que seja encontrado um agrupamento onde a distância seja inferior ou igual ao raio do mesmo, o elemento é inserido neste agrupamento, na Figura 21 é ilustrado este processo.

```

RLCinserção( $L, x$ )
  Se  $L = \{\}$ 
    Retornar  $\{(x, r, \_, \_) \emptyset\}$ 
  Senão
    Seja  $L = \{(c, r, \_, \_) | L'\}$ 
    Se  $d(x, c) \leq r$ 
      Se  $d(x, c) = 0$ 
        Retornar  $L$ 
      Senão
         $(c, r, \_, \_) \leftarrow$  Inserir  $x$  no agrupamento  $(c, r, \_, \_) I$ 
        Retornar  $\{(c, r, \_, \_) | L'\}$ 
    Senão
      Retornar  $\{(c, r, \_, \_) | \text{RLCinserção}(L', x)\}$ 

```

Figura 21. Algoritmo de inserção da “recursive list of clusters”.

A inserção do elemento no agrupamento possui várias situações. Caso o interior deste esteja implementado como “array” e haja espaço para mais um elemento, este é inserido ordenadamente. Mas, se não existir espaço, a implementação do interior do agrupamento é modificada para uma lista de agrupamentos e os elementos prévios do “array” juntamente com o novo elemento são inseridos nesta. Na situação em que o interior já se encontra implementado como uma lista, o algoritmo descrito na Figura 21 é aplicado a esta.

### 5.2.3 Remoção

A remoção de um elemento funciona como o processo inverso à inserção. A lista de agrupamentos é percorrida até encontrar o agrupamento onde o elemento se encontra, Figura 22. No caso de o elemento ser o centro de um agrupamento, este é removido e os elementos do seu interior são reinseridos na sub-lista restante. Se não for encontrado nenhum agrupamento, não existe nada para remover.

```

RLCremoção( $L, x$ )
  Se  $L = \{\}$ 
    Retornar  $L$ 
  Senão
    Seja  $L = \{(c, r, \_, \_) | L'\}$ 
    Se  $d(x, c) \leq r$ 
      Se  $d(x, c) = 0$ 
        Retornar Inserir na sublista restante ( $L, L'$ )
      Senão
         $(c, r, \_, \_) \leftarrow$  remover  $x$  no agrupamento  $(c, r, \_, \_)$ 
        Retornar  $\{(c, r, \_, \_) | L'\}$ 
    Senão
      Retornar  $\{(c, r, \_, \_) | \text{RLCinserção}(L', x)\}$ 

```

Figura 22. Algoritmo de remoção da “recursive list of clusters”.

A remoção do elemento do interior do agrupamento possui várias situações. No caso do interior do agrupamento estar implementado como lista, o algoritmo na Figura 22 é aplicado a esta e se o número de elementos de uma lista atingir o valor da capacidade de um “array” (valor  $\alpha$ ), estes elementos são movidos para um “array”. No caso do interior do agrupamento estar implementado como um “array”, o elemento é simplesmente removido.

#### 5.2.4 Pesquisa

A operação de pesquisa na RLC possui algumas semelhanças com a LC. Dado um par (elemento  $q$ , raio  $r$ ), é percorrida recursivamente a lista de agrupamentos e para cada agrupamento  $(c, r_c)$  é medida a distância  $d(c, q)$ . No entanto é considerado um maior número de situações na relação entre o espaço de pesquisa e o agrupamento de modo a ampliar a eficiência:

- O espaço de pesquisa está contido no agrupamento e abrange o centro deste. Neste caso o centro é adicionado ao conjunto de resposta e o seu interior é processado. Figura 23 a).
- O espaço de pesquisa contém o agrupamento. Neste caso o centro e o interior do agrupamento são adicionados ao conjunto de resposta. Figura 23 b).
- O espaço de pesquisa intersecta o agrupamento contendo o seu centro. Neste caso o centro é adicionado ao conjunto de resposta e o interior do agrupamento é processado. Figura 23 c).

- O espaço de pesquisa está contido no agrupamento mas não abrange o centro deste. Neste caso o seu interior é processado. Figura 23 d).
- O espaço de pesquisa está fora do agrupamento. Neste caso não há nada para realizar. Figura 23 e).
- O espaço de pesquisa intersecta o agrupamento mas não contém o seu centro. Neste caso o interior do agrupamento é processado. Figura 23 f).

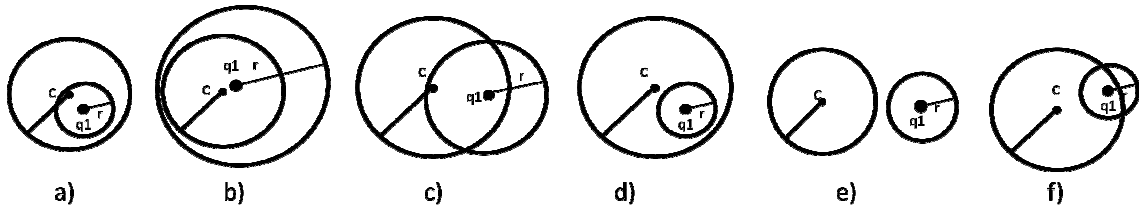


Figura 23. Exemplos de situações de pesquisa.

É importante mencionar que nas situações em que o espaço de pesquisa está contido num agrupamento a iteração da lista é terminada.

No processamento do interior de um agrupamento, se este for implementado por uma “recursive list of clusters” a operação de pesquisa é novamente aplicada a esta. No caso de o agrupamento estar implementado por um “array”, é necessário verificar se os seus elementos estão contidos no espaço de pesquisa. De modo a tornar esta verificação eficiente e evitar uma iteração exaustiva do “array”, a seguinte proposição é usada:

Seja  $x, c, q \in U$  três elementos, e  $r_q$  um número real não negativo tal que  $d(x, c) < d(q, c) - r_q$ . Então  $d(x, q) > r_q$  [60].

Esta proposição indica que existe um limite para a distância entre os elementos do “array” e o centro do agrupamento, para o qual os elementos podem ser descartados sem necessidade de calcular a sua distância ao elemento de pesquisa  $q$ . Por outro lado, a seguinte proposição, indica que o limite  $d(q, c) - r_c$  pode ser usado para determinar se um elemento pertence ao conjunto de resposta sem calcular a sua distância ao elemento de pesquisa  $q$ :

Seja  $x, c, q \in U$  três elementos, e  $r_q$  um número real não negativo tal que  $d(x, c) \leq -(d(q, c) - r_q)$ . Então  $d(x, q) \leq r_q$  [60].

Tendo em conta que os “arrays” estão ordenados por ordem decrescente em relação à distância dos elementos ao centro, estes limites podem ser usados para rejeitar ou aceitar todos os restantes elementos do “array” sem cálculos adicionais. Juntamente com o facto de que cada elemento tem associado a sua sequência  $D$  (que contém as

distâncias aos centros), é possível reduzir drasticamente o número de cálculos da função de distância, tornando ainda mais eficiente a pesquisa.

### 5.3 “Vantage Point Tree” – VPTree

A estrutura de dados métrica “Vantage point tree” – VPTree, introduzida por Peter N. Yianilos [43], baseia-se no método de partição baseado em pivôs. Esta estrutura consiste numa árvore binária dos elementos da base de dados. Cada nó interno é representado por um tuplo  $\langle p, r_p, E, D \rangle$ , onde  $p$  é um elemento da base de dados  $U$  designado por “vantage point” ou pivô,  $r_p$  corresponde ao raio do nó e os conjuntos  $E$  e  $D$  representam as sub-árvores esquerda e direita, respectivamente. Um elemento é adicionado na sub-árvore esquerda se a sua distância ao elemento pivô for inferior ou igual ao raio deste, caso contrário é adicionado à sub-árvore direita, ou seja  $E = \{o \in U - \{p\} \mid d(p, o) \leq r_p\}$  e  $D = \{o \in U - \{p\} \mid d(p, o) > r_p\}$ . Um nó folha é semelhante a um nó interior, no entanto não possui raio e sub-árvores.

Na Figura 24 é exemplificada uma representação gráfica de uma VPTree com dois pivôs  $p_1$  e  $p_2$ , juntamente com uma vista de duas dimensões (parte esquerda).

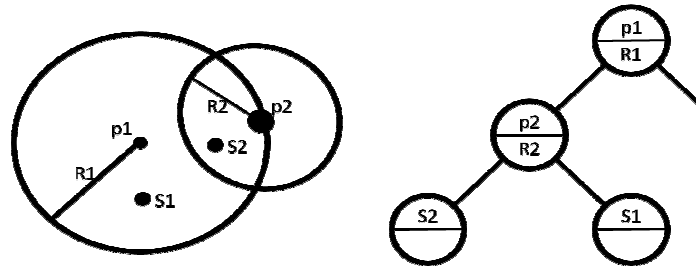


Figura 24. Exemplo de uma VPTree com dois pivôs.

#### 5.3.1 Construção

A “vantage point tree” é construída a partir de um conjunto de elementos inicial através de um processo recursivo. Dada uma base de dados de elementos  $U$ , é seleccionado um determinado elemento  $p$  como pivô e é calculado o seu raio  $r_p$ . Os elementos de  $U - \{p\}$  que possuem uma distância  $d(p, e_{e \in U - \{p\}})$  inferior ao raio  $r_p$  são inseridos na sub-árvore esquerda, caso contrário são inseridos na sub-árvore direita. De seguida o processo é repetido para as sub-árvores de  $p$  até estas serem um conjunto vazio. Na Figura 25 é mostrada uma descrição do algoritmo.

```

VPTree( $U$ )
  Se  $U = \{\}$ 
    Retornar árvore vazia
  Criar (nó)
   $p = \text{Selecionar\_vp}(U)$ 
  Nó.p =  $p$ 
  Nó.raio =  $\text{Mediana}_{u \in U} d(u, p)$ 
   $E = \{u \in U - \{p\} \mid d(p, u) < \text{nó.raio}\}$ 
   $D = \{u \in U - \{p\} \mid d(p, u) > \text{nó.raio}\}$ 
  Nó.E = VPTree( $E$ )
  Nó.D = VPTree( $D$ )
  Retornar nó

```

Figura 25. Algoritmo de construção da VPTree.

O elemento pivô pode ser seleccionado de acordo com os seguintes métodos [43]:

P1: O elemento pivô  $P$  é escolhido aleatoriamente da base de dados;

P2: O elemento pivô  $P$  é calculado de um conjunto de candidatos. Neste método, são obtidos da base de dados dois conjuntos de elementos de forma aleatória. Um é o conjunto de candidatos, o outro, um conjunto que permite a avaliação dos candidatos. Esta avaliação consiste em calcular, para cada candidato, a mediana da distância entre o candidato e os elementos do segundo conjunto, de forma a efectuar o cálculo do segundo momento central (variância). O candidato que apresentar o maior valor de variância é o escolhido. De forma a ilustrar este método, a Figura 26 descreve o algoritmo.

```

SelecionarVP( $U$ )
   $P = \text{Amostra aleatória de } U$ 
   $\text{maiorV} = 0$ 
  Para  $p \in P$ 
     $D = \text{Amostra aleatória de } U$ 
     $m = \text{Mediana}_{d \in D} d(p, d)$ 
     $v = \text{momento2}_{d \in D} (d(p, d) - m)$ 
    Se  $v > \text{maiorV}$ 
       $\text{maiorV} = v$ 
       $\text{melhorP} = p$ 
  Retorna  $\text{melhorP}$ 

```

Figura 26. Algoritmo de selecção de pivôs.

O raio  $r_p$  pode ser seleccionado de acordo com os seguintes métodos [43]:

R1: O valor do raio corresponde à mediana das distâncias entre o elemento pivô  $p$  e os elementos abaixo deste na árvore. Este método permite a construção de uma árvore equilibrada;

R2: O valor do raio corresponde ao valor médio das distâncias entre o elemento pivô  $p$  e os elementos abaixo deste na árvore. Este método foi proposto em [44]. No entanto, o seu uso pode causar uma árvore não equilibrada provocando uma perda de eficiência no algoritmo de pesquisa.

### 5.3.2 Pesquisa

A realização de uma pesquisa consiste em, dado um par (elemento  $q$ , raio  $r$ ), percorrer recursivamente os nós da árvore, começando na raiz até às folhas. Para cada elemento, se a distância  $d(p, q)$  for menor que o raio da pesquisa  $r$ , o elemento  $P$  é adicionado ao conjunto de resposta. Se  $d(p, q) + r \geq r_p$  o processo é repetido para a sub-árvore direita (ver Figura 27, pesquisas  $q1$  e  $q3$ ) e se  $d(p, q) - r \leq r_p$  o processo é aplicado à sub-árvore esquerda (ver Figura 27, pesquisas  $q2$  e  $q3$ ). Importante de mencionar que o processo pode ser aplicado às duas sub-árvores de um nó, se ambas as condições forem satisfeitas (ver Figura 27 pesquisa  $q3$ ). A Figura 27 mostra uma descrição do algoritmo e as representações gráficas dos casos possíveis.

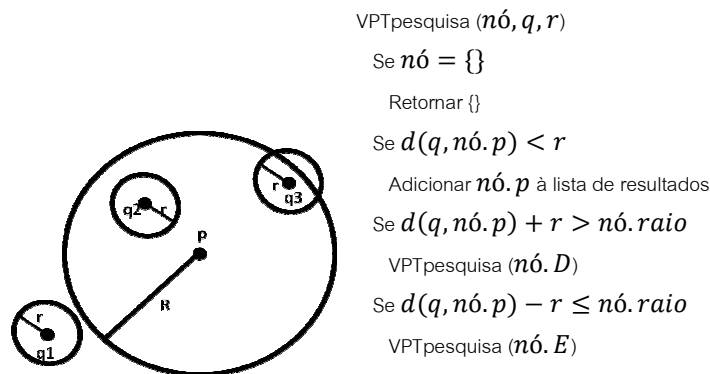


Figura 27. Algoritmo de pesquisa da VPTree.

#### Significado da condição $d(p, q) + r > r_p$ :

A distância  $d(p, q) + r$  representa a maior distância que um possível elemento presente no espaço de pesquisa pode ter do pivô  $p$ . Caso esta distância seja superior ao raio  $r_p$ , então o espaço de pesquisa pode conter elementos que possam estar na sub-



árvore direita, uma vez que esta possui os elementos com uma distância a  $p$  superior a  $r_p$ .

### Significado da condição $d(p, q) - r \leq r_p$ :

A distância  $d(p, q) - r$  representa a menor distância que um possível elemento presente no espaço de pesquisa pode ter do pivô  $p$ . Caso esta distância seja inferior ao raio  $r_p$ , então o espaço de pesquisa pode conter elementos que possam estar na sub-árvore esquerda, uma vez que esta possui os elementos com uma distância a  $p$  inferior a  $r_p$ .

Em [43] e [44] é descrita uma optimização da VPT para melhorar os tempos de pesquisa, reduzindo o número de cálculos de distância. Esta optimização é designada por VP<sup>s</sup>T. Durante a construção da VPT são efectuados cálculos de distância entre cada objecto  $x$  e os elementos pivô, durante o percurso de  $x$  da raiz até à folha em que é inserido. Na VP<sup>s</sup>T estas distâncias  $d(x, p_i)$  são armazenadas no nó  $x$ , ou seja, cada nó contém a mesma informação que os da VPT juntamente com o conjunto  $P_x = \{(p_1, d(x, p_1)), \dots, (p_{\|P_x\|}, d(x, p_{\|P_x\|}))\}$ . A Figura 28 mostra um exemplo de uma representação gráfica da VP<sup>s</sup>T.

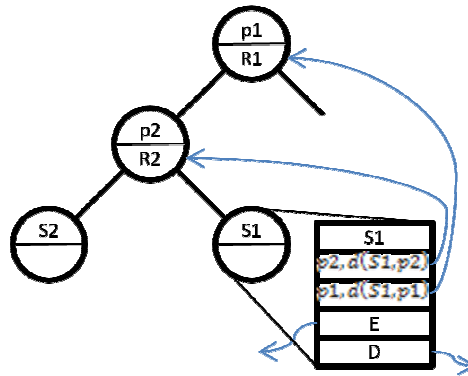


Figura 28. Exemplo de uma VP<sup>s</sup>T com dois pivôs.

No momento da pesquisa de um nó  $x$ , esta informação juntamente com as distâncias  $d(q, p_i)$  anteriormente calculadas, onde  $p_i$  representa um elemento pivô que é antepassado de  $x$ , permitem: descartar o objecto  $x$  se  $|d(q, p_i) - d(x, p_i)| > r$  e incluir  $x$  na lista de resultados se  $d(q, p_i) + d(x, p_i) \leq r$  sem calcular  $d(x, q)$ . Isto é consequência directa de que  $|d(q, p_i) - d(x, p_i)|$  e  $d(q, p_i) + d(x, p_i)$  constituem os limites mínimo e máximo de  $d(x, q)$ , respectivamente.

Justificação: Dadas as distâncias  $d(q, p_i)$  e  $d(x, p_i)$  pela desigualdade triangular verificamos que  $d(x, q) \leq d(q, p_i) + d(x, p_i)$  e que  $d(x, q) + d(x, p_i) \geq d(q, p_i) \Leftrightarrow d(x, q) \geq d(q, p_i) - d(x, p_i)$  e  $d(x, q) + d(q, p_i) \geq d(x, p_i) \Leftrightarrow d(x, q) \geq d(x, p_i) - d(q, p_i)$  logo  $d(x, q) \geq |d(q, p_i) - d(x, p_i)|$ .

#### 5.4 “Geometric Near-Neighbor Access Tree” – GNAT

A estrutura de dados métrica “Geometric Near-Neighbor Access Tree” – GNAT, introduzida por Sergey [45], baseia-se no método de partição baseado em agrupamentos. Um dos objectivos desta estrutura consiste em organizar os elementos da base de dados de uma forma hierárquica por vários domínios de Dirichlet [46] procurando reflectir a disposição geométrica dos elementos no espaço métrico.

Domínio de Dirichlet: Dado um número de pontos  $x_1, \dots, x_m$ , o domínio de Dirichlet de  $x_i$  consiste no conjunto de pontos do espaço que estão mais próximos de  $x_i$  do que os outros pontos  $x_j$  ( $j \neq i$ ).

Como é evidente pelo nome, a GNAT utiliza uma estrutura hierárquica em árvore, em que cada nó desta apresenta um número variável  $m$  de elementos  $< (p_1, D_1), \dots, (p_m, D_m) >$ . Os elementos  $p_i$  correspondem aos centros dos agrupamentos, ou seja, aos pontos  $x_i$  na definição de domínio de Dirichlet. Cada elemento  $p_i$  de um nó, constitui um elemento da base de dados, pelo qual esta é particionada e possui um conjunto  $D_i$  que corresponde aos restantes elementos da base de dados que pertencem ao domínio de Dirichlet de  $p_i$ . Para melhor exemplificar esta estrutura, na Figura 29, é mostrado um exemplo em que os pontos maiores representam os centros do nó de cima e os menores, os centros dos sub-nós. As linhas mais grossas representam as fronteiras dos domínios do nível superior e as linhas mais finas representam as fronteiras dos domínios de nível inferior. Na parte direita é apresentada a representação em árvore dos domínios da parte esquerda.

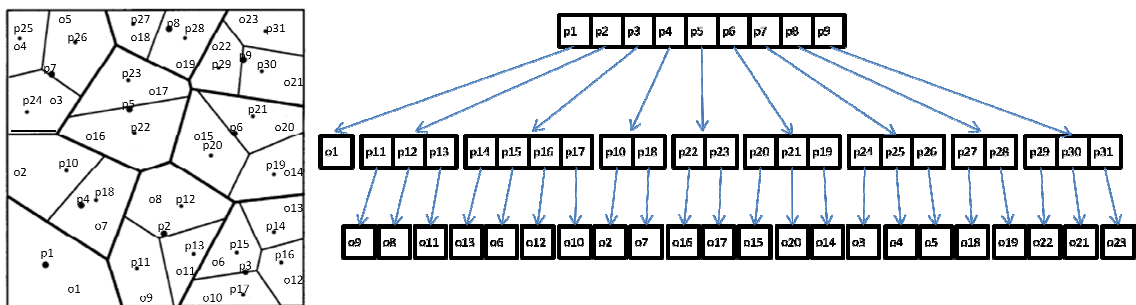


Figura 29. Representação de uma GNAT.

É importante ainda referir que para cada nó é mantida uma tabela  $m \times m$  com os pares de distância mínima e máxima dos elementos  $p_i$  aos subconjuntos  $D_j$ . Formalmente, estas distâncias são definidas por:  $d_{min}(p_i, o_j)$  e  $d_{max}(p_i, o_j)$  para  $o_j \in D_j \cup \{p_j\}$ . Com base nestas distâncias é possível, aquando da pesquisa, descartar domínios que não possuam elementos para a lista de resultados (para mais detalhe ver a secção 5.4.2).

### 5.4.1 Construção

A “Geometric Near-Neighbor Access Tree” é construída a partir de um conjunto de elementos inicial, através de um processo recursivo. Dada uma base de dados de elementos  $U$  são seleccionados  $m$  elementos  $p_i$  de  $U$ , para os quais é definido um conjunto  $D_i$  com os restantes elementos que estejam mais próximos de  $p_i$  do que os outros elementos  $p_j$  ( $j \neq i$ ), ou seja,  $D_i = \{u \in U - \{p_i\} : \forall j \neq i, d(p_i, u) < d(p_j, u)\}$ . Deste modo é formado um nó da árvore e calculada a tabela de pares de distâncias entre cada elemento  $p_i$  e  $D_j$ , de seguida o processo é repetido para os subconjuntos  $D_i$  até que o valor  $m$  seja zero. Na Figura 30 é mostrado uma descrição do algoritmo.

```

GNAT ( $U$ )
   $M = \text{SelecionarM}(U)$ 
  Criar (nó)
  Se  $M = 0$ 
    Retornar  $\text{nó}, \text{centros} = U$ 
   $P = \text{SelecionarCentros}(U)$ 
  Para cada  $p \in P$ 
     $D_i = \{u \in U - \{p_i\} : \forall j \neq i, d(p_i, u) < d(p_j, u)\}$ 
     $t = \text{Para cada par } (p_i, p_j) \text{ calcular } d_{min}(p_i, o_j) \text{ e } d_{max}(p_i, o_j) \text{ para } o \in D_j \cup \{p_j\}$ 
  Nó.centros =  $P$ 
  Nó.tabela =  $t$ 
  Para cada  $p \in P$ 
    Nó.centros, D = GNAT ( $D_i$ )

```

Figura 30. Algoritmo de construção da GNAT.

A escolha dos  $m$  pontos  $p_i$  é realizada em cada nó sobre os elementos ainda por inserir. É seleccionado um conjunto de candidatos com  $3 \times m$  elementos. O primeiro elemento é escolhido de forma aleatória, enquanto os outros são escolhidos de acordo com a distância aos candidatos já escolhidos. Mais concretamente, o primeiro é

escolhido aleatoriamente, o segundo escolhido é o que possui uma maior distância ao primeiro, o terceiro é o que possui uma maior distância aos dois elementos anteriormente escolhidos. Este processo é repetido até se obter  $m$  elementos. Em [44][45] é referido que o valor  $m$  é ajustado conforme o nível na árvore e deve ser proporcional ao número de elementos alocados no nó, no entanto estes detalhes pecam por falta de informação na bibliografia disponível.

### 5.4.2 Pesquisa

A realização de uma pesquisa consiste em, dado um par (elemento  $q$ , raio  $r$ ), percorrer recursivamente os nós da árvore, começando na raiz até às folhas. Em cada nó, é obtido o seu conjunto  $P$  dos elementos  $p_i$  que particionam os elementos da base de dados. Para cada elemento  $p_i$  é calculada a distância deste ao elemento de pesquisa  $d(p_i, q)$ , caso esta seja inferior ou igual ao raio de pesquisa  $d(p_i, q) \leq r$ , o elemento  $p_i$  é adicionado à lista de resultados. Antes de calcular as distâncias de  $q$  aos outros elementos de  $P$ , é possível com base na tabela de distâncias contida em cada nó, descartar domínios de elementos da árvore que não possuam elementos para o conjunto de resposta. Isto é realizado removendo cada elemento  $p_j \in P$  do conjunto  $P$  se  $d(p_i, q) - r > d_{\max}(p_i, o_j)$  (Figura 31, pesquisa q1) ou  $d(p_i, q) + r < d_{\min}(p_i, o_j)$  (Figura 31, pesquisa q2) para  $o \in D_j \cup \{p_j\}$ . Após percorrer os elementos restantes em  $P$ , o processo é repetido para os domínios  $D$  dos elementos presentes em  $P$ . A Figura 31 mostra uma descrição do algoritmo e as representações gráficas dos casos possíveis.

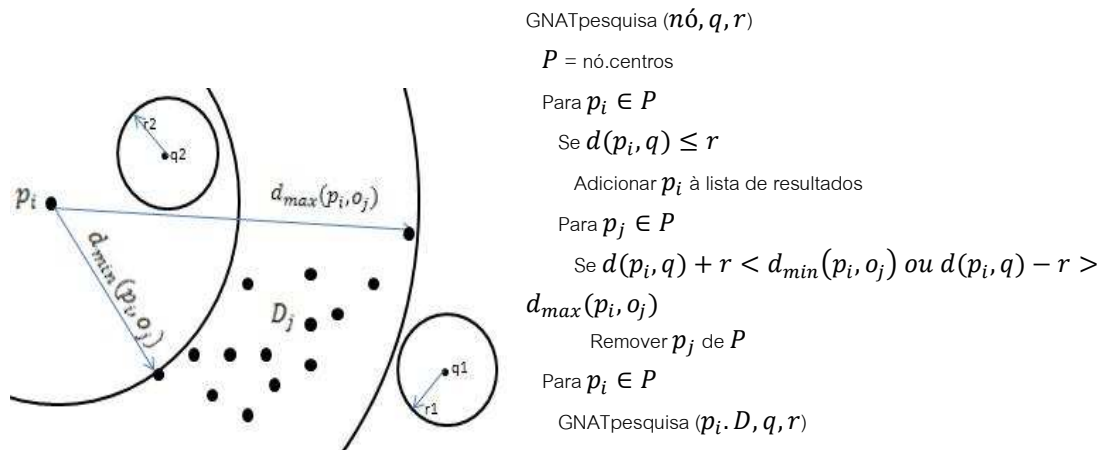


Figura 31. Algoritmo de pesquisa da GNAT.

## 5.5 “Dynamic Spatial Approximation Tree” – DSAT

A estrutura de dados métrica “Dynamic Spatial Approximation Tree” – DSAT, introduzida por D. Arroyuelo, G. Navarro e N. Reyes [48][49], é uma estrutura dinâmica que se baseia no método de partição baseado em agrupamentos.

Esta estrutura consiste numa árvore de aridade  $m$  dos elementos da base de dados. Cada nó interno é representado por um tuplo  $\langle a, t_a, r_a, N_a \rangle$ , onde  $a$  é um elemento da base de dados  $U$ , representando o centro de um agrupamento,  $r_a$  corresponde ao raio do nó, que consiste na distância máxima a qualquer elemento nas suas sub-árvores,  $t_a$  corresponde à estampilha temporal referente ao momento da inserção do nó  $a$  e o conjunto  $N_a$  representa os vizinhos do nó  $a$ . Cada vizinho consiste numa sub-árvore do nó  $a$  e encontram-se organizados por ordem crescente da sua estampilha temporal em  $N_a$ . Considerando o conjunto  $N_a$  como  $\{b_1, \dots, b_k\}$  com  $k \in [0, m]$ , um vizinho  $b_i$  é mais antigo (foi inserido primeiro) que um vizinho  $b_j$  com  $j \in [i, k]$ .

Na Figura 32 é exemplificada uma representação gráfica de uma DSAT com 5 elementos  $a_1, \dots, a_5$  e aridade 2, juntamente com uma vista de duas dimensões (parte esquerda).

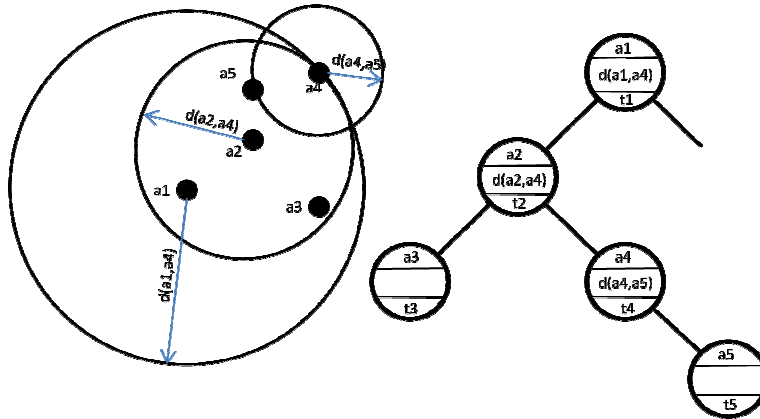


Figura 32. Exemplo de uma DSAT com 5 elementos e aridade 2.

### 5.5.1 Construção

Sendo a DSAT uma estrutura dinâmica, esta é formada por uma sequência de operações de inserção. No entanto, a sua organização depende da ordem dos elementos na inserção, ou seja, com o mesmo conjunto de elementos é possível obter diferentes DSATs se estes forem inseridos com ordem diferente.

### 5.5.2 Inserção

A inserção de um novo elemento  $x$  consiste em percorrer recursivamente a árvore, começando na raiz, e deslocando-se para o vizinho  $b_i$  mais próximo de  $x$ , ou seja,  $d(x, b_i) < d(x, b_j) \wedge d(x, a) > d(x, b_i)$  para  $j \neq i$ . Este deslocamento é repetido até encontrar um nó  $a$  tal que o elemento  $x$  esteja mais próximo de  $a$  do que de qualquer vizinho seu e que a cardinalidade de  $N_a$  seja menor que  $m$ , ou seja,  $\forall b \in N_a, d(x, a) < d(x, b) \wedge \|N_a\| < m$ . Neste caso o elemento  $x$  é inserido como folha do nó  $a$ , no caso em que a cardinalidade de  $N_a$  é igual a  $m$ , o algoritmo é repetido para o vizinho  $b_i$  mais próximo de  $x$ .

Ao longo do percurso da árvore, em cada nó, é verificado se o raio  $r_a$  é menor que a distância  $d(x, a)$ , na situação em que isto se verifica o valor de  $r_a$  é actualizado com o valor  $d(x, a)$ . Um aspecto importante consiste no facto de que um elemento  $a$  ser sempre mais antigo que os seus vizinhos, ou seja  $t_a < t_b$  para  $b \in N_a$ .

A Figura 33 mostra uma descrição do algoritmo de inserção e com base na Figura32 é visível que o elemento  $a_5$  está mais próximo de  $a_2$  mas como a cardinalidade de  $N_{a_2}$  é igual a  $m$  este foi inserido como filho de  $a_4$  por ser o vizinho de  $a_2$  mais próximo.

```

DSATinserção( $a, x$ )
  Se  $a = null$ 
    Retornar  $nó(x)$ 
  Se  $d(x, a) > r_a$ 
     $r_a = d(x, a)$ 
     $d(b_i, x) \leftarrow \min(d(b_i, x) \forall b_i \in N_a)$ 
    Se  $(d(b_i, x) < d(a, x) \text{ ou } \|N_a\| = m)$ 
      DSATinserção( $b_i, x$ )
  Senão
     $N_a \cup \{nó(x)\}$ 

```

Figura 33. Algoritmo de inserção da DSAT.

### 5.5.3 Remoção

Para remover o elemento  $x$ , é primeiro necessário primeiro localizá-lo na árvore. Ao contrário de várias estruturas, para localizar um elemento não é semelhante a simular uma inserção e verificar onde este seria inserido, uma vez que a árvore no momento da remoção pode ser diferente da árvore no instante em que o elemento foi inserido, visto que na inserção são percorridos os nós mais próximos disponíveis. Caso

num momento posterior seja inserido um elemento em que o elemento  $x$  seja mais próximo, esta situação conduziria a percorrer um novo trajecto na árvore.

Em [50], para encontrar um elemento  $x$ , pode-se realizar uma pesquisa por alcance com raio zero, da forma  $(x, 0)$ .

Por uma questão de simplicidade consideremos  $x$  o elemento a remover e  $a$  o seu nó pai. Na situação em que  $x$  está num nó folha a sua remoção é trivial, bastando remover o nó  $x$  de  $N_a$ . No caso em que o nó  $x$  é um nó interior, este é removido de  $N_a$  e é necessário reinserir os seus descendentes de novo na árvore. Por outro lado os elementos de  $N_a$  que são mais novos que  $x$  foram comparados a este para decidir o seu ponto de inserção, o que implica, que na ausência de  $x$  poderiam ser reinseridos numa posição diferente. Deste modo, todos os elementos mais novos que  $x$  na sub-árvore com raiz  $a$  são removidos da árvore, ordenados por ordem crescente da estampilha temporal e reinseridos um por um a partir do nó  $a$ .

#### 5.5.4 Pesquisa

A realização de uma pesquisa consiste em, dado um par (elemento  $q$ , raio  $r$ ), percorrer recursivamente os nós da árvore, começando na raiz até às folhas. Em cada nó  $a$  é calculada a distância  $d(a, q)$  e no caso de  $d(a, q) > r_a + r$  o algoritmo termina para esta sub-árvore, uma vez que esta não possui nenhum elemento da lista de resultados.

Justificação: Sendo  $r_a$  a distância de  $a$  ao elemento  $x$  mais afastado presente nos descendentes de  $a$ , temos que  $\forall y \in \text{Descendentes}_a \ d(a, y) \leq d(a, x)$  e  $d(a, q) > d(a, x) + r$ . E  $d(a, q) > d(a, x) + r \Leftrightarrow d(a, q) - d(a, x) > r$ , da desigualdade triangular  $d(q, x) + d(a, q) \geq d(a, x) \Leftrightarrow d(q, x) \geq d(a, q) - d(a, x)$ , logo temos  $d(q, x) > r$ .

Na situação em que  $d(a, q) \leq r_a + r$ , se  $d(q, a) \leq r$  o elemento  $a$  é adicionado à lista de resultados. De seguida, é necessário pesquisar nos vizinhos de  $a$  os que possam conter elementos no espaço de pesquisa, considerando  $N_a$  como  $\{b_1, \dots, b_k\}$ , a sub-árvore  $b_i$  é pesquisada se  $d(q, b_i) \leq \min(d(q, b_1), \dots, d(q, b_{i-1})) + 2r$ , isto deve-se ao facto que algum elemento  $f$  que seja mais próximo de  $b_{i+j}$  com  $j \in [1, m - i]$ , tenha sido inserido em  $b_i$  por causa de o elemento  $b_{i+j}$  ainda não existir, assim só é necessário pesquisar os sub-nós mais novos se estes tiverem uma distância a  $q$  menor

que a distância mínima entre  $q$  e um sub-nó mais antigo. O parâmetro  $2r$  corresponde à tolerância do raio de pesquisa.

Um aspecto que permite reduzir o trabalho efectuado nos sub-nós é a informação temporal presente na estampilha de cada nó. Caso se verifique  $d(q, b_i) > d(q, b_{i+j}) + 2r$  é ainda necessário pesquisar o sub-nó  $b_i$  por este ser mais antigo, uma vez que os elementos inseridos antes de  $b_{i+j}$  podem ter sido inseridos em  $b_i$ , mas apenas é necessário considerar os elementos com estampilha menor que a de  $b_{i+j}$ . Visto que os nós descendentes têm uma estampilha maior que a dos pais, quando um nó  $n$  tiver uma estampilha maior que a de  $b_{i+j}$ ,  $n$  e todos os seus sub-nós podem ser descartados. A Figura 34 mostra uma descrição do algoritmo de pesquisa que na sua invocação recebe um valor de estampilha temporal  $t = \infty$ .

```

DSATpesquisa( $a, x, r, t$ )
  Se  $t_a < t \wedge d(a, q) \leq r_a + r$ 
    Se  $d(a, q) \leq r$ 
      Adicionar  $a$  à lista de resultados
     $d_{min} = \infty$ 
    Para  $b_i \in N_a$ 
      Se  $d(b_i, q) \leq d_{min} + 2r$ 
         $k \leftarrow \min(j > i, d(b_i, q) > d(b_j, q) + 2r)$ 
        DSATpesquisa( $b_i, q, r, t_{b_k}$ )
       $d_{min} = \min\{d_{min}, d(b_i, a)\}$ 

```

Figura 34. Algoritmo de pesquisa da DSAT.

## 5.6 “Hybrid Dynamic Spatial Approximation Tree” – H-DSAT

A estrutura de dados métrica “Hybrid Dynamic Spatial Approximation Tree” – H-DSAT, introduzida por D. Arroyuelo, G. Navarro e N. Reyes [48][49], é uma estrutura dinâmica que resulta da DSAT pela adição de pivôs, permitindo melhorar os tempos de pesquisa pela redução de cálculos de distâncias.

Esta estrutura de dados apresenta uma composição dos elementos da base de dados em árvore, semelhante à DSAT. Em cada nó  $x$ , para além dos elementos referidos na DSAT, é mantido um conjunto  $P_x$  das distâncias aos elementos pivôs associados a  $x$ .

Existem duas versões designadas por H-DSAT1 e H-DSAT2, onde a principal diferença entre ambas consiste no critério de selecção dos pivôs. No entanto em ambas as versões os pivôs são seleccionados aquando da inserção de elemento.



Na versão “Hybrid Dynamic Spatial Approximation Tree 1” – H-DSAT1 os ascendentes de um nó  $x$  constituem o conjunto de pivôs de  $x$ , que está definido por  $P_x = \{(p_i, d(x, p_i)), p_i \in \mathcal{A}_x\}$  sendo  $\mathcal{A}_x$  o conjunto de antepassados de  $x \in U$ .

Na versão “Hybrid Dynamic Spatial Approximation Tree 2” – H-DSAT2 são ainda considerados mais pivôs que na H-DSAT1. Para além dos antepassados do nó  $x$ , são considerados como pivôs os elementos de  $N_{p_i}$  mais antigos que o antepassado  $p_{i+1}$ . Formalmente, seja  $x \in U$  e  $\mathcal{A}_x = \{p_1, \dots, p_k\}$  o conjunto dos seus antepassados, onde  $p_i$  é o antepassado do nível  $i$ , temos que  $(b, d(x, b)) \in P_x$  se  $b \in \mathcal{A}_x$  ou  $p_i, p_{i+1} \in \mathcal{A}_x \wedge b \in N_{p_i} \wedge t_b < t_{p_{i+1}}$ . Repare que  $p_{i+1} \in N_{p_i}$ .

### 5.6.1 Construção

A construção da H-DSAT é semelhante à da DSAT, ou seja, através de uma sequência de operações de inserção.

### 5.6.2 Inserção

A inserção de um elemento  $x$  é realizada da mesma forma que na DSAT, no entanto, no início da inserção o conjunto  $P_x$  é inicializado a  $\emptyset$ , e à medida que os vários nós são percorridos são adicionados a  $P_x$  os pares  $(p_i, d(x, p_i))$  onde  $p_i$  é um elemento pivô de  $x$ . A diferença da inserção nas duas versões, H-DSAT1 e H-DSAT2, consiste no critério de selecção dos pivôs, descritos anteriormente.

### 5.6.3 Remoção

A remoção de um elemento  $x$  é realizada de forma semelhante que na DSAT, no entanto é necessário remover os elementos  $(p_i, d(x, p_i))$  de  $P_x$  dos objectos reinseridos, para os quais  $p_i$  já não é um antepassado de  $x$ .

### 5.6.4 Pesquisa

A realização de uma pesquisa, dado um par (elemento  $q$ , raio  $r$ ), é semelhante à realizada na DSAT, no entanto, usando a informação dos pivôs presente em cada nó  $a$ , é possível verificar se a sub-árvore  $a$  possui elementos que intersectam o espaço de

pesquisa sem necessidade de calcular a distância  $d(q, a)$ . Isto é conseguido pelo cálculo do limite mínimo de  $d(q, a)$  que é designado por  $\mathcal{D}(q, a)$ .

Calculo de  $\mathcal{D}(q, a)$ : No momento da pesquisa no nó  $a$  já temos as distâncias  $d(q, p_i)$  onde  $p_i \in P_a$  e em  $a$  temos  $P_a = \{(p_1, d(a, p_1)), \dots, (p_{\|P_a\|}, d(a, p_{\|P_a\|}))\}$ , deste modo pela desigualdade triangular  $d(a, q) + d(q, p_i) \geq d(a, p_i) \Leftrightarrow d(a, q) \geq d(a, p_i) - d(q, p_i)$  temos que o limite mínimo  $\mathcal{D}(q, a) = d(a, p_i) - d(q, p_i)$  e  $\mathcal{D}(q, a) \leq d(a, q)$ .

Com base no algoritmo de pesquisa da DSAT se  $\mathcal{D}(a, q) > r_a + r$  é verificado então  $d(a, q) > r_a + r$  (explicado em 5.5.4) também se verifica, podendo descartar a pesquisa no nó  $a$  sem calcular  $d(q, a)$ . No caso em que  $d(a, q) \leq r_a + r$  é necessário pesquisar os nós  $x \in N_a$ . Neste ponto já foi calculado as distâncias  $d(q, p_i)$  onde  $p_i \in \mathcal{A}_x$ , repare que isto inclui  $d(q, a)$ . Como cada elemento  $x$  possui o conjunto  $P_x = \{(p_1, d(x, p_1)), \dots, (p_{\|P_x\|}, d(x, p_{\|P_x\|}))\}$  é possível calcular  $\mathcal{D}(q, x)$  de acordo com a explicação anterior e descartar os nós  $x \in N_a$  em que  $\mathcal{D}(q, x) > r_x + r$  se verifica.

## 5.7 “Linear Approximating Eliminating Search Algorithm” – LAESA

A estrutura de dados métrica “Linear Approximating Eliminating Search Algorithm” – LAESA, introduzida por M. Luisa e J. Oncina [51], baseia-se no método de partição baseado em pivôs.

Esta estrutura utiliza  $k$  pivôs para particionar os  $n$  elementos da base de dados e consiste numa matriz  $n \times k$ , onde cada célula  $i, j$  contém a distância do pivô  $i$  ao elemento  $j$ , ou seja,  $d(k_i, n_j)$ . Cada pivô constitui um elemento da base de dados, seleccionado aquando da construção da estrutura de dados. Na Figura 35 é exemplificada uma representação gráfica da LAESA.

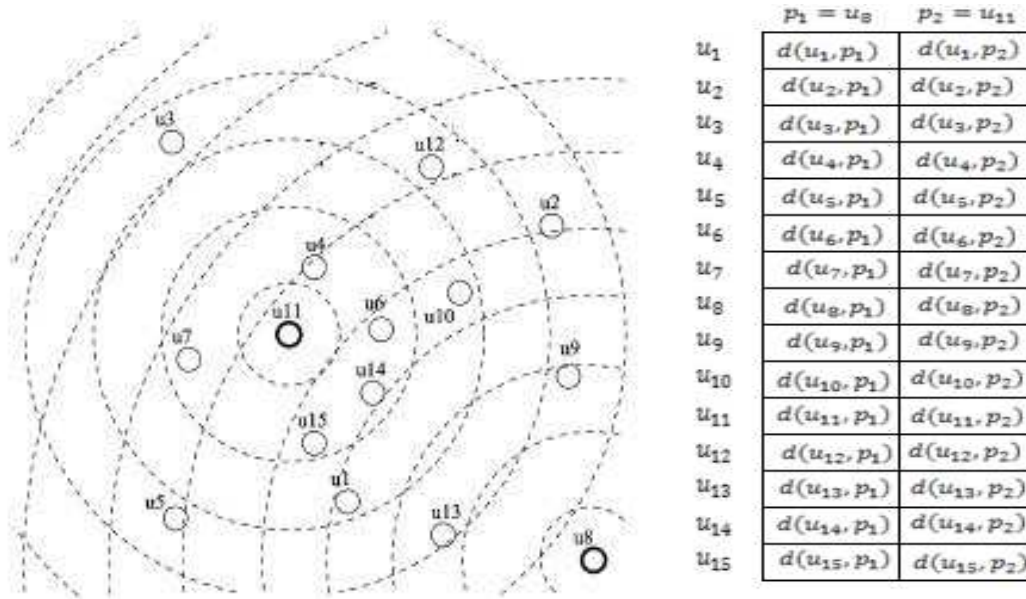


Figura 35. Exemplo de uma LAESA com 15 elementos e pivôs  $u_8$  e  $u_{11}$ .

### 5.7.1 Construção

A “Linear Approximating Eliminating Search Algorithm” é construída a partir de um conjunto de elementos inicial, do qual são seleccionados  $k$  elementos como pivôs. Para cada pivô é calculada a sua distância a cada elemento do conjunto inicial, guardando-a na respectiva célula da matriz  $L$ . A selecção dos pivôs é realizada de modo a tentar obter pivôs que estejam mais afastados entre si.

De forma a tornar a construção o mais eficiente possível, a selecção dos pivôs e os cálculos das distâncias, entre estes e todos os elementos da base de dados, são realizados em simultâneo, da seguinte forma. O primeiro pivô é escolhido de forma aleatória. Após a selecção de um pivô  $p$ , é calculada a distância  $d(p, u)$  deste a todos os  $n$  elementos da base de dados  $U$ . Para cada elemento  $u$  da base de dados é mantido um acumulador  $A[u]$  que é incrementado com a distância do elemento  $u$  ao último pivô escolhido, ou seja,  $d(p, u)$ . O próximo elemento pivô é escolhido a partir do acumulador cujo valor é maior, ou seja, é seleccionado o elemento que possui a maior distância aos pivôs anteriormente seleccionados. Este processo é repetido até se obter um conjunto  $P$  de pivôs com dimensão  $k$ . A Figura 36 mostra uma descrição breve deste algoritmo.

```

LAESA ( $U$ )
 $p = \text{elem\_aleatório}(U)$ 
 $P = \{p\}$ 
 $A = [0]$ 
Enquanto  $\|P\| < m$ 
     $max = 0$ 
     $pp = p$ 
    Para  $u \in U - P$ 
         $L[p, u] = d(p, u)$ 
         $A[u] = A[u] + L[p, u]$ 
        Se  $A[u] > max$ 
             $pp = u$ 
             $max = A[u]$ 
     $P = P \cup \{pp\}$ 

```

Figura 36. Algoritmo de construção da LAESA.

Relativamente à escolha do valor  $k$ , com base em [52][53], temos  $k \geq \dim + 1$  em que  $\dim$  representa o número de dimensões dos elementos da base de dados.

### 5.7.2 Pesquisa

Seja  $S = \{s_1, \dots, s_2\}$  um conjunto igual a  $U$  e  $P = \{p_1, \dots, p_k\}$  o conjunto de pivôs.

A realização de uma pesquisa consiste em, dado um par (elemento  $q$ , raio  $r$ ), iterar sobre  $P$  e em cada iteração, calcular a distância entre  $q$  e  $p_i$ , ou seja,  $d(q, p_i)$ . Após o cálculo de  $d(q, p_i)$ , para cada elemento  $s_i \in S$  se  $|d(q, p_i) - d(s_1, p_i)| > r$  se verificar podemos descartar  $s_i$  do conjunto  $S$ . Esta iteração termina quando  $S = \{\}$  ou tenha sido iterado todo o conjunto  $P$ .

Justificação: Dadas as distâncias  $d(q, p_i)$  e  $d(s_i, p_i)$  (presente na matriz) pela desigualdade triangular verificamos que  $d(s_i, q) + d(s_i, p_i) \geq d(q, p_i) \Leftrightarrow d(s_i, q) \geq d(q, p_i) - d(s_i, p_i)$  e  $d(s_i, q) + d(q, p_i) \geq d(s_i, p_i) \Leftrightarrow d(s_i, q) \geq d(s_i, p_i) - d(q, p_i)$  logo  $d(s_i, q) \geq |d(q, p_i) - d(s_i, p_i)|$ . Sendo  $|d(q, p_i) - d(s_i, p_i)|$  o limite mínimo para  $d(s_i, q)$ , se  $d(s_i, q) > r$  podemos concluir que  $s_i$  não pertence ao espaço de pesquisa.

Na situação em que  $S = \{\}$  a pesquisa termina verificando-se que não houve resultados. Caso contrário, para cada elemento  $s_i \in S$  é calculada directamente a distância de  $q$  a  $s_i$ , ou seja,  $d(q, s_i)$ . Se  $d(q, s_i) \leq r$  se verificar, então  $s_i$  é adicionado a lista de resultados.

## 5.8 Conclusões e contributos

Neste capítulo foram descritas oito estruturas de dados métricas (incluindo as duas variantes H-DSAT1 e H-DSAT2) dando ênfase aos seus processos de construção e de pesquisa por alcance. O conjunto de estruturas de dados apresentado é composto pela LC, RLC, VPtree, GNAT, DSAT, H-DSAT1, H-DSAT2 e LAESA. Todas estas estruturas foram seleccionadas para avaliar a sua aplicabilidade e eficiência em aplicações de pesquisa de música semelhante.

Das estruturas de dados apresentadas é possível observar que apenas um subconjunto das estruturas é dinâmico, sendo constituído pela RLC, DSAT, H-DSAT1 e H-DSAT2.

Durante o estudo realizado foi encontrado um trabalho na área de MIR que empregou a VPtree e mediu a sua performance [1]. No entanto, não foram encontrados trabalhos que estabelecessem uma comparação dos resultados de diferentes estruturas de dados métricas na área de MIR. Sendo apenas, encontrados dois trabalhos no domínio dos dicionários de línguas [57][74] que comparam os resultados de diferentes estruturas de dados métricas. Em [57] foram usadas as estruturas GNAT, H-DSAT1, LAESA, LC, RLC e VPtree. Enquanto em [74] foram usadas as estruturas RLC e VPtree.



## **Parte II – Implementação e Avaliação**





## 6. Pacote de estruturas de dados métricas

Neste capítulo é descrita a implementação do pacote de estruturas de dados métricas implementadas e que foram usadas na avaliação. No capítulo 5 foram apresentadas e descritas diversas estruturas de dados, sendo as seguintes escolhidas para avaliação: LAESA, VPTree, DSAT, H-DSAT1, H-DSAT2, LC, GNAT e RLC.

A implementação das estruturas de dados métricas foi realizada na linguagem JAVA [67], e baseadas nas descrições apresentadas no capítulo 5. Esta implementação foi realizada juntamente com o colega Pedro Chambel<sup>2</sup>, sendo estabelecido como meta o desenvolvimento de um pacote de estruturas de dados métricas que possa ser utilizado em diferentes aplicações.

Foi implementado um pacote de estruturas totalmente genérico, que permite o uso de diferentes tipos de objectos e medidas de distância. A Figura 37 ilustra a estrutura do pacote desenvolvido numa ferramenta de modelagem UML [73].

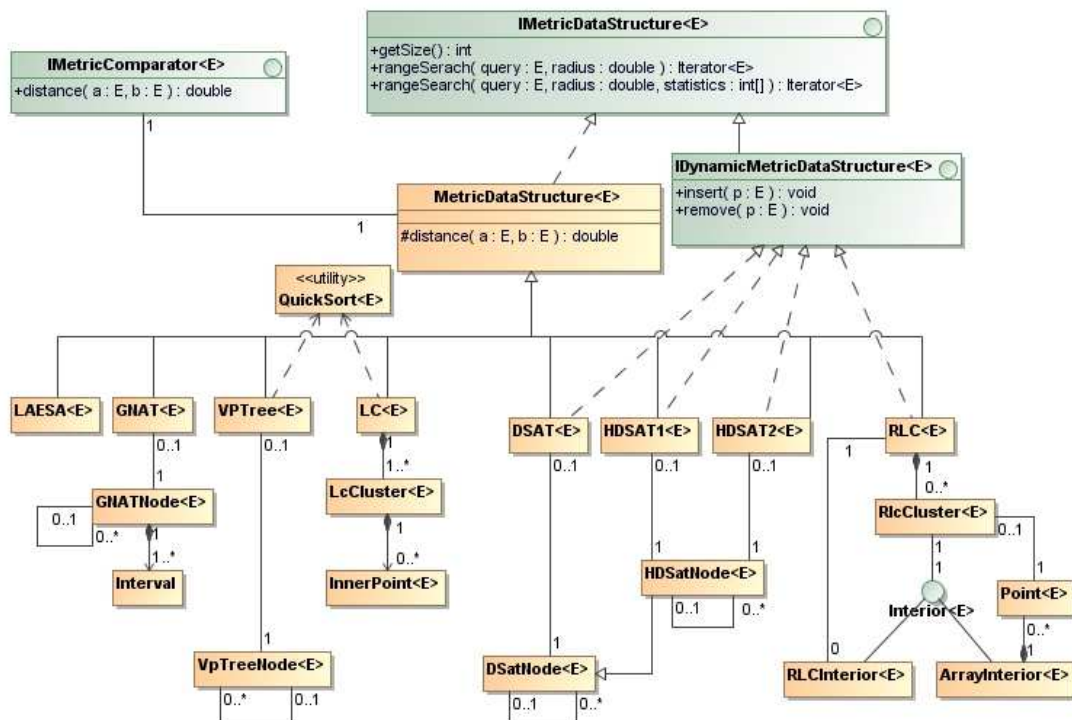


Figura 37. Diagrama de classes do pacote de estruturas de dados métricas.

<sup>2</sup> Aluno de mestrado a realizar a tese intitulada “Pesquisa de Imagens de Rosto” orientada pela Prof. Dra. Fernanda Barbosa.

Este capítulo está organizado em três secções. Na primeira secção proceder-se-á à descrição das interfaces utilizadas. Numa segunda secção serão definidas as classes do pacote. E por último, é apresentada a conclusão.

## 6.1 Descrição das interfaces

As interfaces “IMetricDataStructure” e “IDynamicMetricDataStructure” definem as estruturas de dados métricas presentes neste pacote de software. Todas estas estruturas implementam a interface “IMetricDataStructure” e apenas as estruturas dinâmicas implementam a interface “IDynamicMetricDataStructure”.

A interface “IMetricDataStructure” disponibiliza duas funções de pesquisa por alcance. Ambas produzem o mesmo resultado, no entanto, a função “rangeSearch(query : E, radius : double, statistics : int[] ) : Iterator<E>” recebe como argumento um “array” de inteiros, com o objectivo de armazenar as estatísticas referentes à pesquisa. Na última versão deste pacote de software, na posição zero do “array” é possível obter o número de cálculos de distância efectuados durante a execução da pesquisa. Embora esta seja a única estatística extraída da pesquisa, foi estabelecido o uso de um “array” com o objectivo de facilitar a adição de outras estatísticas numa futura versão.

A interface “IDynamicMetricDataStructure” disponibiliza as funções “insert( p : E) : void” e “remove( p : E) : void” que permitem respectivamente a inserção de um novo elemento e a remoção de um dado elemento.

Pelo facto das estruturas de dados métricas se basearem no conceito de espaço métrico todas elas necessitam duma métrica, a qual é representada na interface “IMetricComparator” e define a função de cálculo de distância entre dois objectos. Para definir uma medida de distância a utilizar no pacote de estruturas, é necessário que esta implemente a interface “IMetricComparator”. Relativamente aos objectos do espaço métrico não existe qualquer restrição, apenas é necessário que estes sejam implementados de modo a funcionar com o comparador.

## 6.2 Descrição das classes

A classe abstracta “MetricDataStructure” é estendida por todas as estruturas de dados métricas e contém a secção de código comum entre estas estruturas, isto permite uma factorização do código desenvolvido e uma maior legibilidade. Esta classe contém

a variável que representa o comparador e a função “distance( a : E, b : E ) : double” que retorna o valor de distância entre os elementos a e b.

De seguida são ilustradas as classes que representam cada estrutura de dados métrica e são mencionados alguns detalhes da implementação de cada estrutura.

## 6.2.1 LAESA

A Figura 38 ilustra a implementação da estrutura de dados métrica LAESA. Esta estrutura apenas possui um construtor, que recebe como argumentos a métrica (comp), o número de pivôs (nrPivots) e o conjunto de elementos (elems) do espaço métrico.

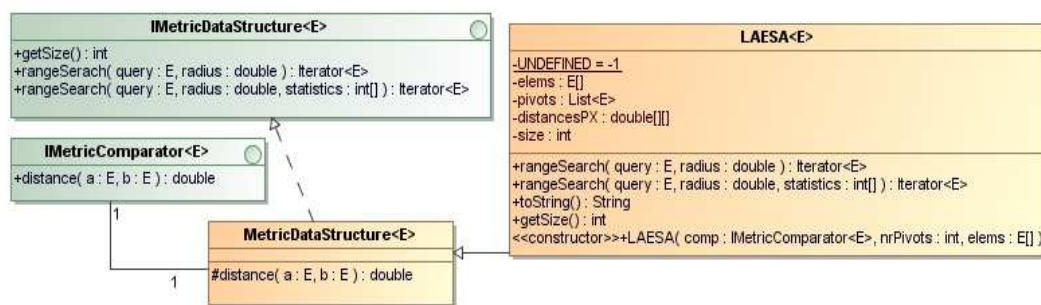


Figura 38. Implementação da estrutura LAESA.

## 6.2.2 VPtree

A Figura 39 ilustra a implementação da estrutura de dados métrica VPtree. Esta estrutura apenas possui um construtor, que recebe como argumentos a métrica (comp) e o conjunto de elementos (elems) que definem o espaço métrico. Na secção 5.3.1 foram indicadas algumas opções para a selecção do elemento pivô e do seu raio, sendo utilizado na implementação os métodos P1 (o elemento pivô  $p$  é escolhido aleatoriamente da base de dados) e R1 (o valor do raio corresponde à mediana das distâncias entre o elemento pivô  $p$  e os elementos abaixo deste na árvore).

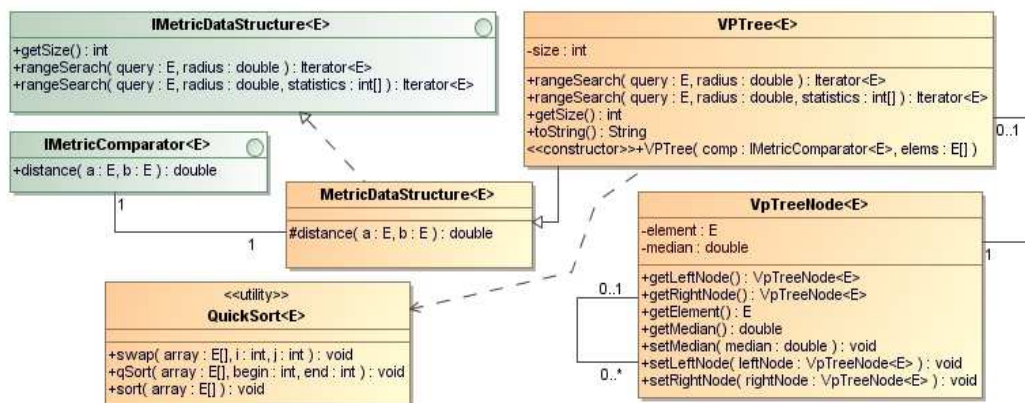


Figura 39. Implementação da estrutura VPtree.

### 6.2.3 DSAT, H-DSAT1 e H-DSAT2

A Figura 40 ilustra a implementação das estruturas de dados métricas DSAT, H-DSAT1 e H-DSAT2. Estas estruturas possuem dois construtores, o primeiro recebe como argumentos a métrica (comp), a aridade da árvore (maxArity) e o conjunto de elementos (elems) que definem o espaço métrico. O segundo construtor apenas recebe como argumento a métrica (comp) e a aridade da árvore (maxArity), sendo a estrutura construída por sucessivas operações de inserção.

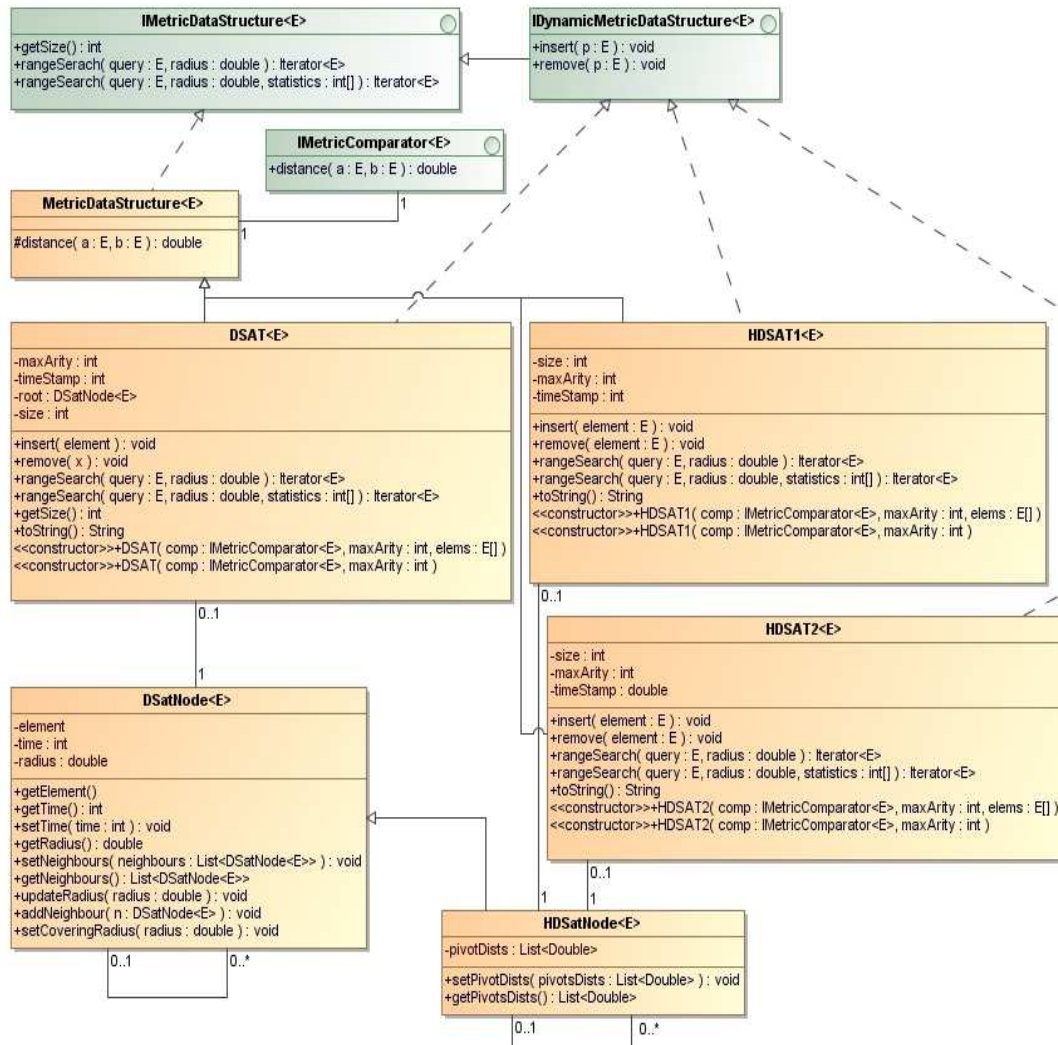


Figura 40. Implementação das estruturas DSAT, H-DSAT1 e H-DSAT2.

### 6.2.4 LC

A Figura 41 ilustra a implementação da estrutura de dados métrica LC. Esta estrutura possui dois construtores, onde se verifica que cada construtor representa um método de selecção do raio dos agrupamentos. O primeiro construtor representa o método A1 (partições com raio fixo: todos os agrupamentos possuem o mesmo raio) e

recebe como argumentos a métrica (comp), o raio dos agrupamentos (radius) e o conjunto de elementos (elems) do espaço métrico. O segundo construtor representa o método A2 (partições de tamanho fixo: todos os agrupamentos possuem o mesmo número de elementos e o raio é definido de acordo com a distância do elemento mais afastado do centro) e recebe como argumentos a métrica (comp), o tamanho dos agrupamentos (clSize) e o conjunto de elementos (elems) do espaço métrico. Relativamente às heurísticas apresentadas na secção 5.1.1 para a selecção dos centros dos agrupamentos, foi implementada a heurística H4 (o elemento que minimiza a soma das distâncias aos centros dos agrupamentos anteriores) para o construtor de partições com raio fixo, e a heurística H5 (o elemento que maximiza a soma das distâncias aos centros dos agrupamentos anteriores) para o construtor de partições de tamanho fixo. A escolha das heurísticas foi baseada nos resultados obtidos em [41] pelo autor da estrutura LC.

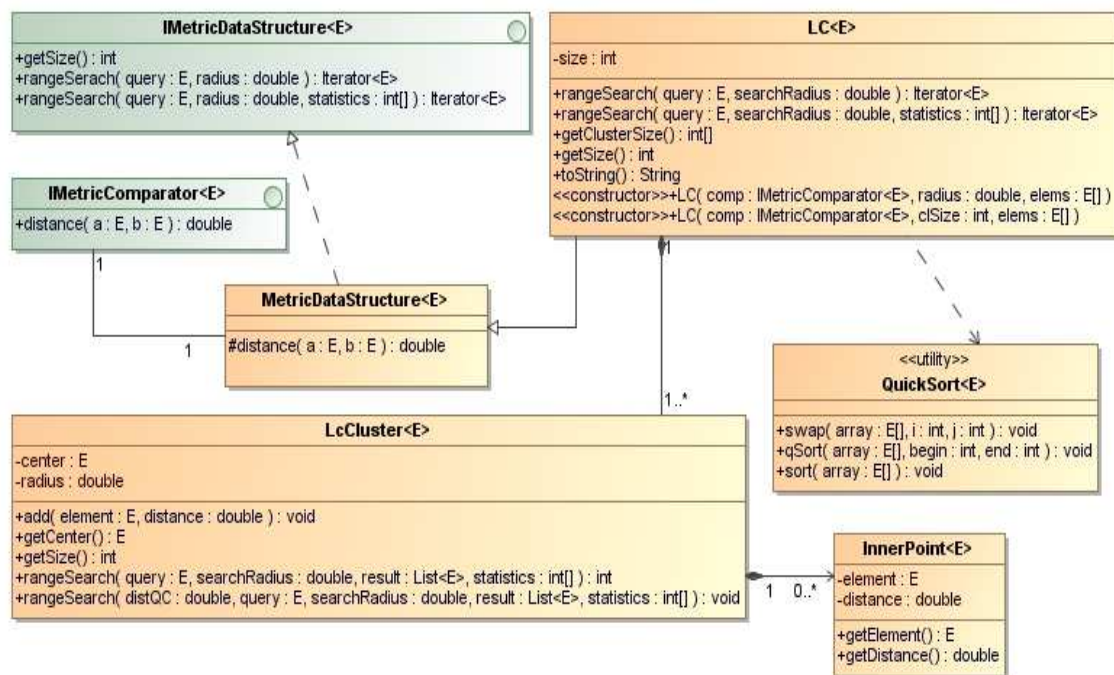


Figura 41. Implementação da estrutura LC.

## 6.2.5 GNAT

A Figura 42 ilustra a implementação da estrutura de dados métrica GNAT. Esta estrutura apenas possui um construtor, que recebe como argumentos a métrica (comp), o “degree” (número de centros em cada nó) e o conjunto de elementos (elems) do espaço métrico. O método implementado para a selecção dos  $m$  centros de cada nó foi



anteriormente descrito na secção 5.4.1, embora não seja realizado qualquer ajuste ao valor  $m$  conforme o nível na árvore e a proporção de elementos alocados no nó.

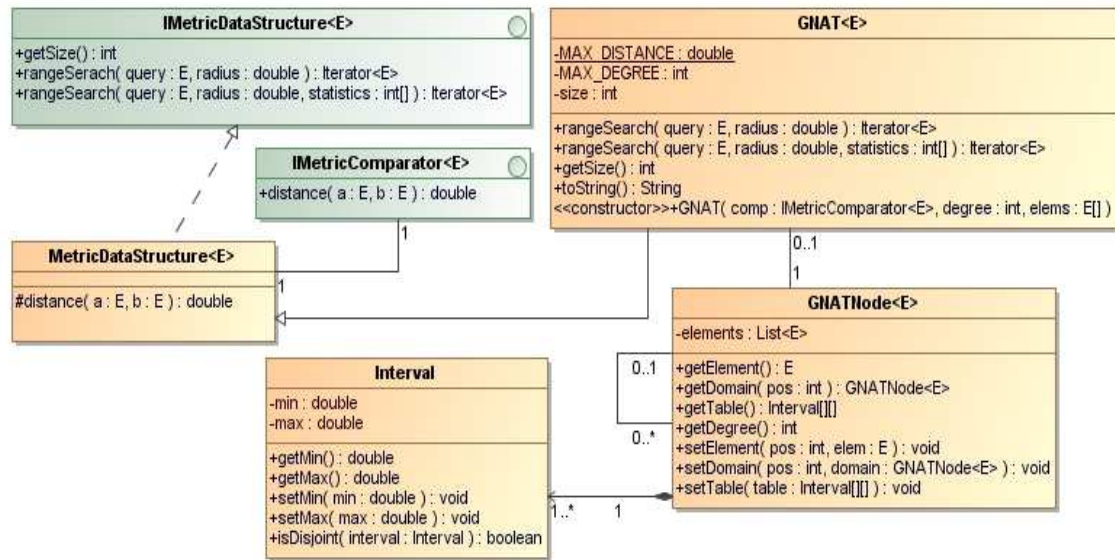


Figura 42. Implementação da estrutura GNAT.

## 6.2.6 RLC

A Figura 43 ilustra a implementação da estrutura de dados métrica RLC. Esta estrutura possui dois construtores, o primeiro recebe como argumentos a métrica (comp), o raio dos agrupamentos (radius), a capacidade da RLC (arrayCapacity) e o conjunto de elementos (elems) do espaço métrico. O segundo construtor apenas recebe como argumentos a métrica (comp), o raio dos agrupamentos (radius) e a capacidade da RLC (arrayCapacity), sendo a estrutura construída por sucessivas operações de inserção.

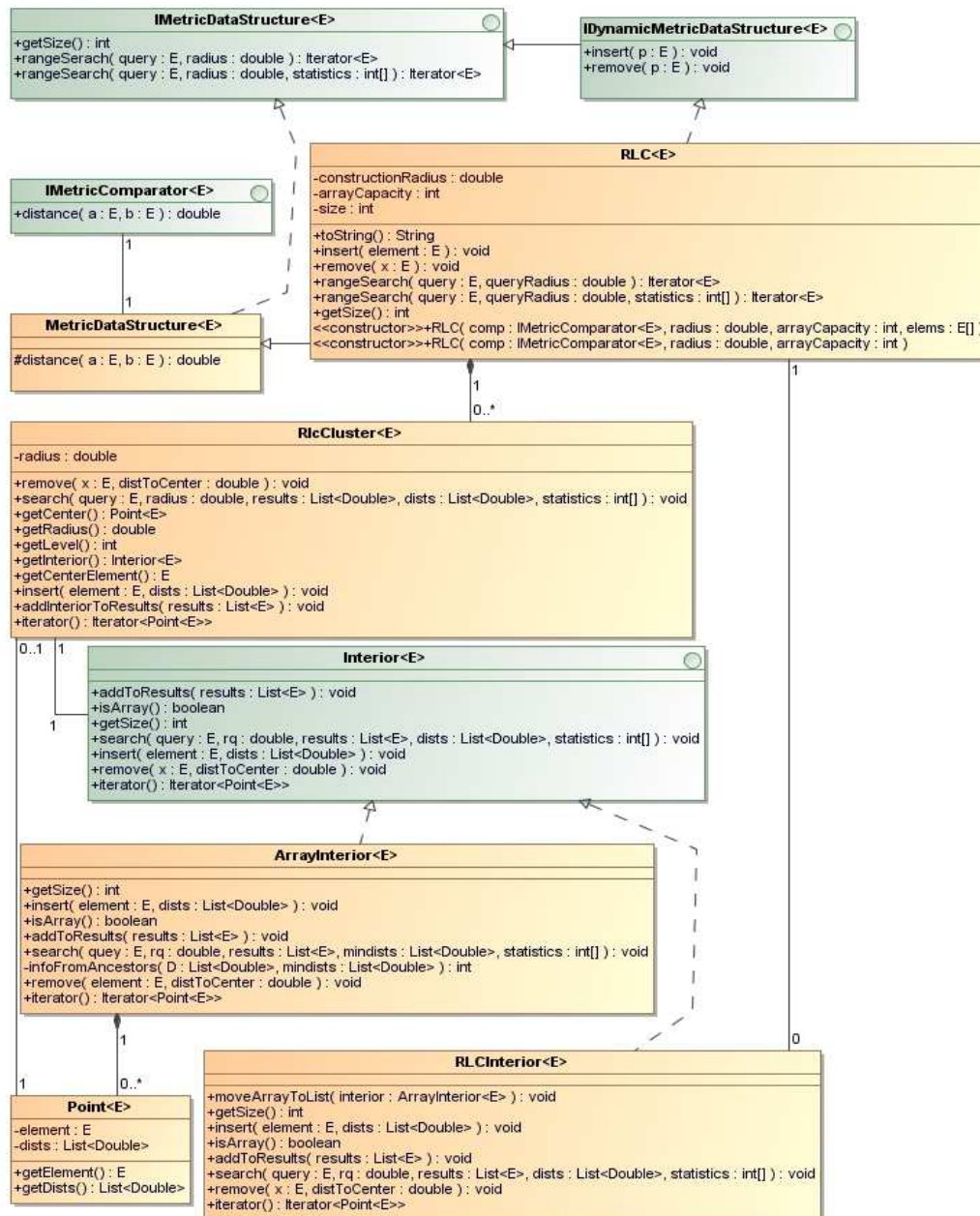


Figura 43. Implementação da estrutura RLC.

## 6.3 Conclusões

Com o pacote desenvolvido de estruturas de dados métricas é possível utilizar estas estruturas de dados em diferentes aplicações. Este pacote é totalmente genérico, permitindo o uso de diferentes espaços métricos e possibilitando a sua reutilização para futuros projectos.

No âmbito desta tese todas as estruturas de dados serão utilizadas no capítulo 8, de modo a avaliar a pesquisa por alcance.

Durante a implementação do pacote procurou-se favorecer o requisito da modificabilidade. Isto permitiu obter uma estrutura que possibilita facilmente a sua expansão, nomeadamente a adição de novas estruturas de dados métricas.



## 7. Espaços métricos

Neste capítulo são descritos os espaços métricos seleccionados que foram usados para avaliar a pesquisa por alcance nas estruturas de dados métricas. Com base nas representações da música apresentadas no capítulo 3 e nas funções de semelhança definidas no capítulo 4, foram implementados três espaços métricos. Tendo em conta a caracterização da música em várias dimensões, descrita no capítulo 3, é possível situar os espaços métricos em duas dimensões: a melodia e o timbre. Para a dimensão melodia foi implementado um espaço métrico que utiliza a representação de intervalos melódicos (secção 3.1.1), sendo a métrica uma variante da distância de edição (secção 4.1). Relativamente à dimensão timbre foram implementados dois espaços métricos que se baseiam na representação da música com MFCCs (secção 3.2.1) e nas distâncias de Manhattan e euclidiana (secção 4.2) para determinar a similaridade.

Para as duas dimensões definidas, foram utilizadas diferentes bases de dados de música, uma vez que, não foi possível obter as mesmas músicas no formato Midi e Wave. Embora se tenha ponderado a conversão de formatos, sabemos que os ficheiros simbólicos não possuem informação relativa ao timbre [15], e para além disso, as músicas em formato Wave eram polifónicas, pelo que a sua conversão para Midi resultaria em ficheiros Midi polifónicos.

É de notar que ainda que foi possível definir espaços métricos no domínio da música, a sua avaliação, no que diz respeito à sua aplicabilidade nas aplicações existentes, é um processo muito complicado. Para esta avaliação foram desenvolvidos protótipos que possibilitam uma avaliação futura destas métricas. Para além disso, foi pedida ajuda ao Professor Doutor Eduardo Lopes<sup>3</sup>.

Este capítulo está estruturado em três secções. As duas primeiras descrevem os espaços métricos das dimensões do timbre e da melodia, respectivamente. A última apresenta as conclusões.

---

<sup>3</sup> Professor Auxiliar do departamento de música da Universidade de Évora.

## 7.1 Timbre

Os espaços métricos definidos nesta dimensão da música procuram capturar informação relativa ao espectro do sinal de áudio de uma música. Conceptualmente, isto corresponde ao timbre de uma música, ou seja, o tipo de instrumentos presentes e a existência, ou não, de voz.

Na dimensão timbre, foram implementados dois espaços métricos. Ambos os espaços utilizam a mesma representação de músicas, sendo a sua diferença a função de distância usada. Cada música será caracterizada por uma assinatura que é calculada com base no seu conteúdo musical e representa a música na dimensão do timbre. As métricas usadas para o critério de semelhança são a Manhattan e a euclidiana.

Deste modo, nesta secção, é descrito o processo de cálculo da assinatura de uma música, a base de dados utilizada para ambos os espaços e as medidas de semelhança.

### 7.1.1 Processamento da assinatura

O processo de cálculo da assinatura de uma música está ilustrado na Figura 44 (adaptada de [61]).

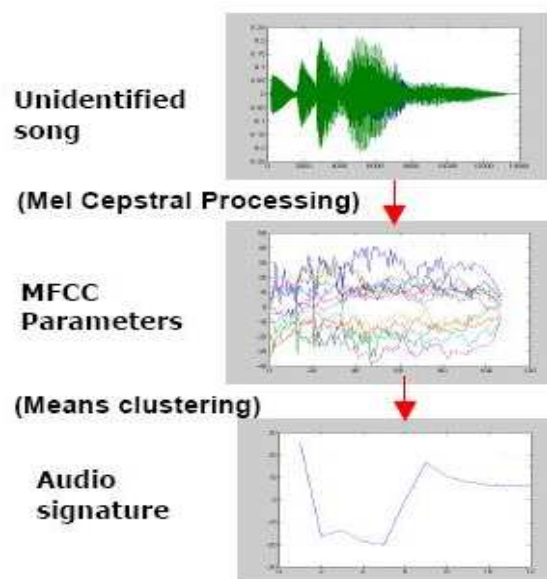


Figura 44. Processo de cálculo da assinatura de uma música com base nos MFCCs.

Este método é semelhante ao utilizado no estudo de Alfie Tan Kok Leong em [61], descrito anteriormente na secção 4.2. De uma forma breve, este processo pode ser descrito em quatro passos:

- Passo 1: Ler o ficheiro Wave de música a processar;

- Passo 2: Dividir o sinal de áudio em frames de 26,5 milissegundos com sobreposição de 15,6 milissegundos entre frames consecutivas;
- Passo 3: Computar os MFCCs para cada frame. Semelhante ao estudo em [61], apenas foram usados treze coeficientes. Esta parametrização deve-se ao facto que em [61] o autor refere a pesquisa de Beth Logan [18] onde se concluiu que um número superior a treze não melhora a performance/precisão da semelhança e que um número inferior resulta numa estimação imperfeita do sinal de áudio;
- Passo 4: Computar o algoritmo “k-means” com um agrupamento (“simple-means”) de modo a obter uma descrição compacta do sinal de áudio, sendo esta usada como assinatura de uma música.

O processo de geração dos MFCCs foi implementado na linguagem MATLAB [66] usando funções proporcionadas pela VoiceBox [62], uma biblioteca de funções para processamento de discurso<sup>4</sup>.

Para produzir uma assinatura de uma música é necessário executar três funções, pela seguinte ordem, “wavread”, “melcepst” e por último “kmeans”. A função “wavread” é a função standard do MATLAB para a leitura de ficheiros de Wave, tomando como argumento um ficheiro Wave e retornando como resultado um vector (“array”) de dados representando o ficheiro Wave. “Melcepst” recebe como argumento o vector que representa o Wave e produz um conjunto de MFCCs. Por último a função “kmeans” produz a assinatura final de uma música criando uma representação compacta do conjunto de MFCCs, sendo esta assinatura um vector de números reais com dimensão treze. A Figura 45 ilustra o processo de criação da assinatura para a música “awake.wav”.

```
%Step 1
wave = wavread('awake.wav');
%Step 2 and 3
mfccSet = melcepst(wave);
%Step 4
signature = kmeans(mfccSet,1);
```

Figura 45. Exemplo de geração da assinatura da música “awake.wav”.

<sup>4</sup> Esta biblioteca de funções é disponibilizada segundo os termos da “GNU Public License” [63].

### 7.1.2 Base de dados de músicas

De modo a poder usar estes espaços métricos, foi criada uma colecção<sup>5</sup> de músicas no formato Wave com um “sampling rate” de 11025 Hertz. Esta colecção é constituída por 250 músicas completas (com a duração original), onde cada uma tem associado o nome da música, nome do artista, nome do álbum e género. Com base em [64], a nossa base de dados é composta pelos seguintes géneros musicais: “alternative”, “alternative rock”, “ambient”, “dance”, “electronic”, “goth and dark era”, “gothic rock”, “gothic metal”, “hard rock”, “instrumental”, “metal”, “new age”, “nu metal”, “punk”, “punk rock”, “rock”, “soundtrack”, “symphonic metal” e “trance”. O facto da base de dados ser composta por uma larga variedade de géneros permite testar a precisão da medida de semelhança, apesar de este teste constituir um processo subjectivo e estar para além dos objectivos desta dissertação. Em [61] foi utilizada uma base de dados diferente, no entanto esta consiste numa colecção de música privada, possuindo direitos de autor. Por outro lado, cada música apenas possuía um minuto de áudio com um “sampling rate” de 16000 Hertz.

### 7.1.3 Medida de semelhança

A medida de semelhança entre músicas é calculada com base na assinatura do espectro de cada música. Esta assinatura é obtida de acordo com a descrição da secção 7.1.1 e apenas necessita de ser calculada uma única vez, sendo armazenada e associada à respectiva música por razões de performance. Posteriormente as assinaturas das músicas são comparadas usando as distâncias de Manhattan e euclidiana.

As métricas da família  $L_p$  (secção 4.2) permitem calcular a distância entre sequências com o mesmo comprimento  $d$ . Neste caso, as assinaturas das músicas são vectores de números reais de dimensão 13, logo é possível utilizar estas métricas.

De modo a caracterizar o espaço métrico foram calculadas a média, a variância e o desvio padrão das distâncias entre os vários elementos da base de dados. Na Tabela I estão apresentados valores para os dois espaços métricos descritos nesta secção.

---

<sup>5</sup> Durante o período de pesquisa e de elaboração dos espaços métricos não foi encontrada nenhuma colecção online/pública de músicas no formato wave, deste modo foi criada uma colecção privada de músicas.

Tabela I. Propriedades dos espaços métricos.

Métrica	Manhattan	Euclidiana
Média	5,791928	2,413186
Variância	5,075894	1,244004
Desvio padrão	2,252975	1,115349

Embora o objectivo não seja a avaliação da precisão da medida de semelhança, uma vez que esse estudo já foi realizado em [61, secção 9], foi implementado em JAVA um protótipo que serve para a avaliação destas métricas. Neste protótipo é possível seleccionar uma dada música e listar todas as músicas da base de dados ordenadas por ordem decrescente de semelhança com a música seleccionada. Nesta listagem é apresentado também o valor da métrica usada. A Figura 46 ilustra a interface desse protótipo, utilizando a base de dados de músicas e permitindo o uso das duas distâncias semelhança.

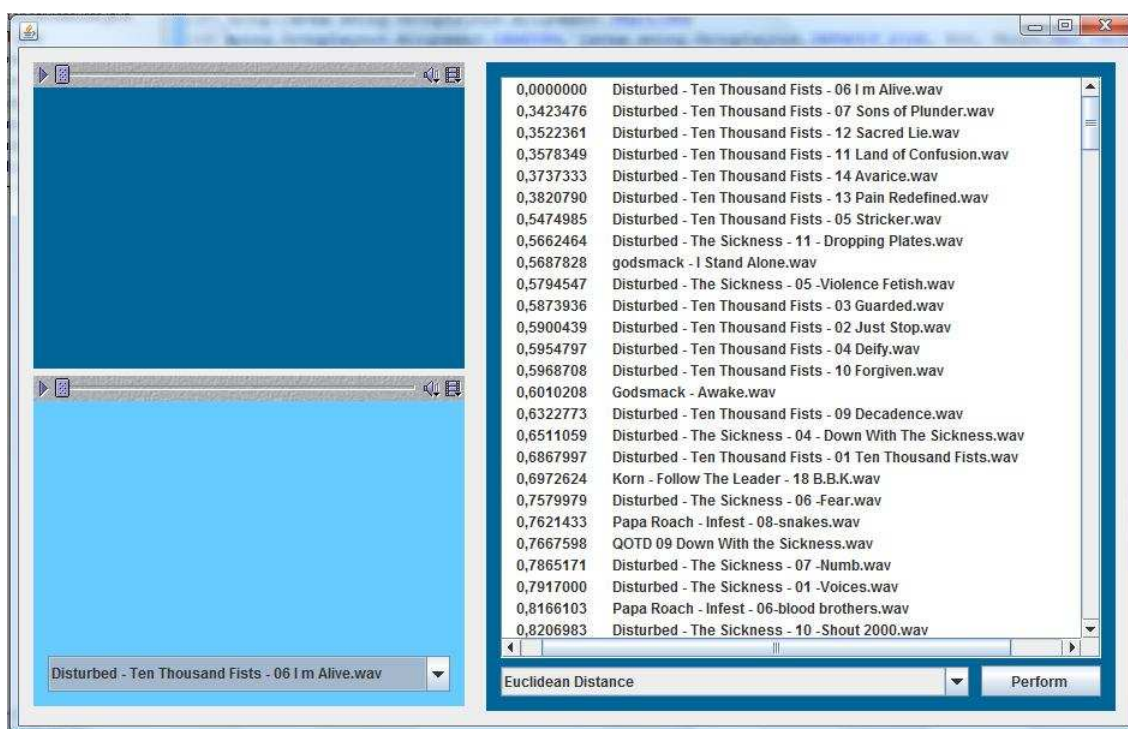


Figura 46. Protótipo da similaridade na dimensão timbre.

Com este protótipo foram realizadas algumas pesquisas e pode-se concluir que as medidas de semelhança aqui implementadas apresentam resultados satisfatórios para as pesquisas realizadas. Esta afirmação tem como base, o facto de entre os 10 primeiros resultados mais próximos de cada pesquisa, estarem presentes em média 5 músicas do mesmo álbum e/ou artista, existindo algumas músicas de outros artistas e álbuns, que de acordo com a classificação de género em [64], apresentam o mesmo género musical.

Para além destas pesquisas realizadas, este protótipo foi utilizado pelo Professor Doutor Eduardo Lopes, o qual concluiu que as medidas de semelhança representam de facto uma medida de proximidade na dimensão do timbre.

7.2 Melodia

O espaço métrico definido nesta dimensão da música procura transcrever e capturar informação relativa às notas musicais do(s) instrumento(s) ou voz de uma música. Conceptualmente, isto corresponde à melodia de uma música, ou seja, à sequência de pitches e durações transmitida por um instrumento ou pela voz de um artista.

Este espaço métrico utiliza a notação de sucessões de intervalos melódicos (secção 3.1.1) como forma de representar uma música e emprega uma variante da distância de edição como medida de similaridade (secção 4.1).

Nesta secção, é descrito o processo de cálculo da assinatura de uma música, a base de dados utilizada e por último, a medida de semelhança.

7.2.1 Processamento da assinatura

O processo de cálculo da assinatura de uma música monofónica é ilustrado na Figura 47 (adaptada de [3]).

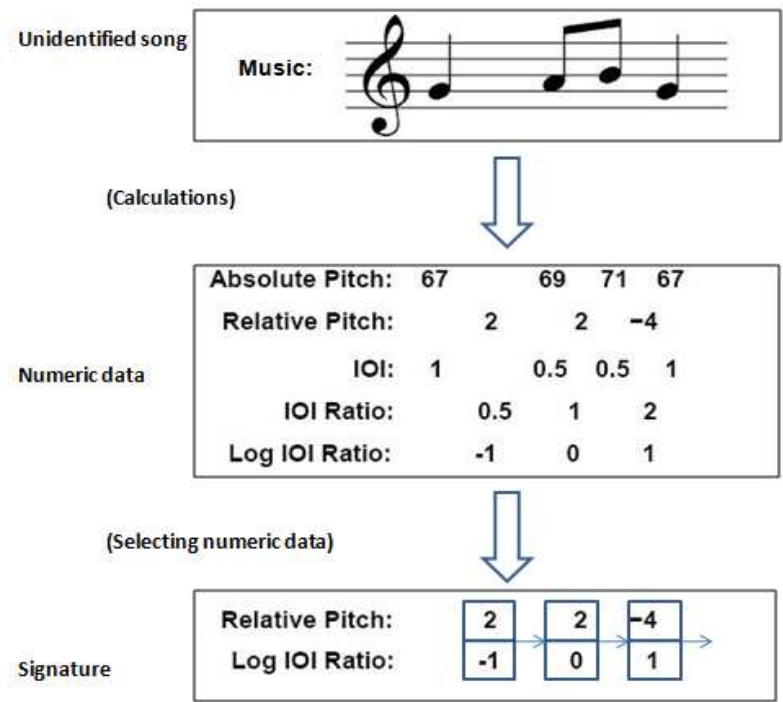


Figura 47. Processo de cálculo da assinatura de uma música com base nos intervalos melódicos.

Este método é semelhante ao utilizado nos projectos de investigação [1], [2] e [3], onde a melodia de uma música é transcrita numa sucessão de intervalos melódicos. Estes intervalos foram descritos anteriormente na secção 3.1.1. Assim sendo, de uma forma breve, este processo pode ser descrito em 3 passos:

- Passo 1: Ler o ficheiro Midi de música a processar;
- Passo 2: Iterar a sequência de notas da melodia calculando os valores descritos em 3.1.1;
- Passo 3: Seleccionar os valores dos pitches relativos e dos LIRs, formando uma sequência de intervalos melódicos que constituem a assinatura de uma música.

O processo de geração das sequências de intervalos melódicos foi implementado na linguagem JAVA usando funções proporcionadas pela jMusic<sup>6</sup> [65], a qual é uma biblioteca de ferramentas para processamento de áudio em JAVA.

Para produzir uma assinatura de uma música é necessário criar um objecto “Score” e inicializá-lo com a respectiva música. Após este primeiro passo, a biblioteca jMusic disponibiliza métodos para explorar a estrutura do “Score” com o objectivo de iterar a sucessão de notas do “track” que representa a melodia. Durante a iteração das notas, são disponibilizadas funções para obter o tempo de início, duração e pitch de cada nota, de modo a calcular os valores requeridos. Finalmente, com base nos valores obtidos é possível construir a assinatura de uma música, baseada na sequência de intervalos melódicos. A Figura 48 ilustra o processo de criação da assinatura para a música “awake.mid”.

---

<sup>6</sup> Esta biblioteca de funções é disponibilizada segundo os termos da “GNU Public License” [68].

```

//step 1
Score s = new Score();
Read.midi(s, "awake.mid");
//notes iteration
Enumeration enum1 = s.getPartList().elements();
while(enum1.hasMoreElements()){
    Part part = (Part) enum1.nextElement();
    Enumeration enum2 = part.getPhraseList().elements();
    while(enum2.hasMoreElements()){
        Phrase phrase = (Phrase) enum2.nextElement();
        double startTime = phrase.getStartTime();
        Enumeration enum3 = phrase.getNoteList().elements();
        while(enum3.hasMoreElements()){
            Note note = (Note) enum3.nextElement();
            if (note.getPitch() != JMC.REST) {
                // start time
                startTime.add(startTime);
                // pitch
                absPitch.add(note.getPitch());
                // duration
                duration.add(note.getDuration());
            }
            startTime += note.getDuration();
        }
    }
}
//initialized data, step 2
relPitch = new int[absPitch.size() - 1];
double[] ioi = new double[startTime.size()];
lir = new double[absPitch.size() - 1];
for(int i = 0; i < absPitch.size(); i++) {
    //relative pitch
    if(i < absPitch.size()-1)
        relPitch[i] = absPitch.get(i+1) - absPitch.get(i);
    //ioi
    if(i < startTime.size()-1)
        ioi[i] = startTime.get(i+1) - startTime.get(i);
    else //last note
        ioi[i] = duration.get(i);
    //lir
    double lirn = 0;
    if(i > 0) {
        lirn = ioi[i] / ioi[i - 1];
        lir[i - 1] = Math.round(Math.log(lirn) / Math.log(2));
    }
}
//step 3
this.melody = new Melody(this.relPitch, this.lir);

```

Figura 48. Exemplo de geração da assinatura da música “awake.mid”.

## 7.2.2 Base de dados de músicas

De forma a poder usar um espaço métrico na dimensão da melodia, foi utilizada a colecção de músicas folclóricas do projecto ESAC<sup>7</sup> [69][70]. Estas músicas têm uma codificação própria da melodia, baseada na notação chinesa JIANPU [70 secção 24] e cada música é identificada por uma designação que possui um código e o número do item, por exemplo A0001. A base de dados é composta por cerca de vinte mil músicas, no entanto nesta dissertação apenas foram consideradas 5223 melodias, as quais eram as únicas que se encontravam disponíveis em [69]. Esta colecção contém música folclórica da China e de vários países europeus, nomeadamente, da Alemanha, Polónia e Áustria.

<sup>7</sup> O projecto ESAC tem como principal motivação a análise e recuperação de informação de música e disponibiliza gratuitamente a sua colectânea de músicas ao público interessado nesta área. Sendo utilizada na conferência de avaliação de sistemas MIR, a MIREX, em várias tarefas de investigação entre as quais, a similaridade melódica [71] assume também um objectivo deste espaço métrico.



Tendo em mente o processo de cálculo das assinaturas das músicas, foi utilizada uma pequena aplicação em JAVA que permite a transcrição do formato simbólico usado na base de dados para o formato Midi (secção 2.3.1). A aplicação que permite a conversão para Midi monofónico foi desenvolvida no projecto de investigação [72] de Carlos Esteban Gómez Chacón e facultada directamente pelo autor para este estudo.

### 7.2.3 Medida de semelhança

A medida de semelhança é calculada com base na assinatura da melodia de cada música. Esta assinatura é obtida de acordo com a descrição da secção 7.2.1 e apenas necessita de ser calculada uma única vez, sendo armazenada e associada à respectiva música por razões de performance. De seguida, é possível medir a semelhança entre músicas, usando uma variante da distância de edição apresentada em 4.1.

A variante da distância de edição foi implementada em JAVA, usando uma técnica de programação dinâmica, tendo em mente a eficiência do sistema. A similaridade melódica  $D(R, S)$  entre duas melodias  $R = \langle I_{R1}, \dots, I_{Rn} \rangle$  e  $S = \langle I_{S1}, \dots, I_{Sm} \rangle$  é computada pelo preenchimento da matriz  $m_{1..n \ 1..m}$ . Cada entrada  $m_{ij}$  representa a similaridade melódica entre dois prefixos  $\langle I_{R1}, \dots, I_{Ri} \rangle$  e  $\langle I_{S1}, \dots, I_{Sj} \rangle$ . O preenchimento de  $m_{ij}$  é ilustrado na Figura 49, onde  $d$  representa a distância entre dois intervalos melódicos (secção 4.1) com  $a = 0,5$  e  $b = 0,5$ . Os valores dos coeficientes  $a$  e  $b$  não são referidos nos artigos [1] e [3], onde foi usada, mas foram facultados pelos autores para a realização deste estudo.

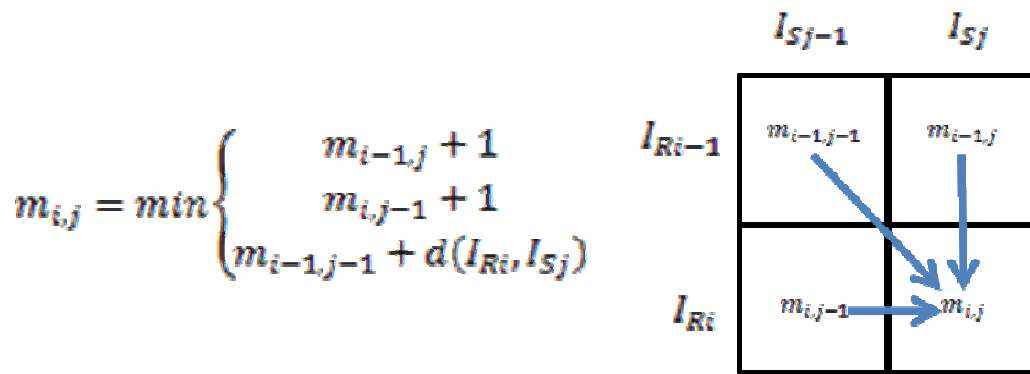


Figura 49. Padrão de cálculo da matriz de distância.

No fim do preenchimento da matriz  $m$ , a entrada  $m_{nm}$  contém o valor da similaridade entre as duas melodias  $R$  e  $S$ .

No trabalho [3], onde se usou esta medida, os autores comentaram que para além de captar a proximidade melódica, também era tomado em conta o ritmo. Esta informação referente ao ritmo é tomada em conta aquando da codificação da informação referente à duração das notas, o logaritmo do rácio do “interonset interval” (LIR).

De modo a caracterizar o espaço métrico foram calculadas a média, a variância e o desvio padrão das distâncias entre os vários elementos da base de dados. Na Tabela II são indicadas as propriedades do espaço métrico descrito nesta secção.

Tabela II. Propriedades do espaço métrico da melodia.

Métrica	Distância de edição
Média	53,229792
Variância	328,736741
Desvio Padrao	18,131099

Para proceder à avaliação desta medida de distância foi implementado um protótipo em JAVA que permite realizar pesquisas por semelhança. Este protótipo foi utilizado pelo Professor Doutor Eduardo Lopes, o qual concluiu que a medida de semelhança representa de facto uma medida de proximidade na dimensão da melodia, mas no ritmo não é notória. A Figura 50 ilustra este protótipo da avaliação de similaridade, utilizando a base de dados de músicas e a distância de edição descrita.

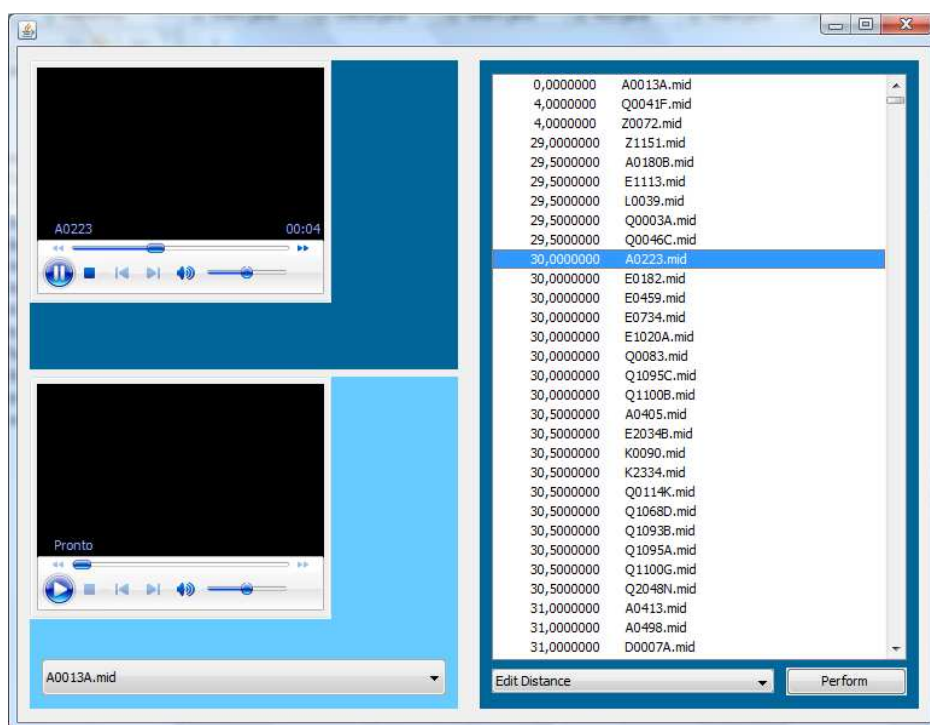


Figura 50. Protótipo da similaridade na dimensão melodia.

### 7.3 Conclusões

Neste capítulo foram descritos três espaços métricos que permitem medir a similaridade de músicas baseada somente no seu conteúdo musical no que diz respeito às dimensões de timbre e melodia. O que nos leva a concluir em parte que é possível usar estruturas de dados métricas nas pesquisas por semelhança. Para cada dimensão da música, timbre e melodia, foram utilizadas diferentes bases de dados de músicas, onde cada base de dados possui um diferente formato de ficheiro de música, Wave e Midi. Estas diferenças tornam impossível estabelecer comparações entre os espaços das dimensões timbre e melodia.

A avaliação das medidas de semelhança aqui descritas é um processo muito complexo, e em verdade está fora do âmbito desta dissertação. No entanto, é importante realçar os seguintes aspectos:

- Foram desenvolvidos protótipos que poderão ser utilizados no futuro para uma avaliação mais formal destas medidas de semelhança;
- Com base na opinião do Professor Doutor Eduardo Lopes, a percepção musical não só se faz a nível fisiológico/cognitivo, como também baseado em premissas culturais (formas apreendidas). A percentagem de cada uma destas perspectivas está longe de ser definida, pois é muito difícil quantificar o que é culturalmente apreendido. Por outro lado, se uma máquina é capaz de extrair características (refinadas) de timbre ou de melodia e medir a sua semelhança, o sistema cognitivo do ser humano poderá não estar tão desenvolvido para as captar.

Por último, segundo o Professor Eduardo Lopes tendo em conta que o ritmo musical manifesta-se transculturalmente como dos parâmetros musicais mais importantes (se não o mais) no processo da cognição musical, talvez este seja o parâmetro a formalizar com maior detalhe num sistema de geração de “playlists”. Tendo como base o que foi descrito nos trabalhos [3] e [61], os quais usaram estas medidas e nas opiniões do Professor Doutor Eduardo Lopes, descritas anteriormente podemos, no âmbito desta tese, ter como premissa que as métricas aqui descritas são satisfatórias no sentido de medir semelhanças nas respectivas dimensões. Assim sendo, pode-se concluir que é possível definir espaços métricos no domínio da música e portanto usar estruturas de dados métricas nas pesquisas por semelhança.



## **8. Pesquisa por alcance em estruturas de dados métricas**

Neste capítulo é descrita a avaliação da eficiência da pesquisa por alcance nas estruturas de dados métricas nos vários espaços métricos implementados no capítulo 7 sobre as respectivas bases de dados (coleções de música). Nesta avaliação, são utilizados os dois espaços métricos da dimensão timbre (secção 7.1) e o espaço da dimensão melodia (secção 7.2), onde cada qual possui a sua respectiva coleção de música, com o objectivo de avaliar a pesquisa por alcance nas estruturas de dados métricas: LAESA, VPTree, DSAT, H-DSAT1, H-DSAT2, LC, GNAT e RLC.

De modo a avaliar a pesquisa por alcance com as diferentes estruturas de dados métricas é necessário que esta avaliação seja “representativa” e justa para todas as estruturas de dados. Deste modo existem diferentes aspectos a ter em conta: (1) o número de pesquisas a realizar; (2) quais as pesquisas a realizar no que diz respeito aos elementos e aos raios de pesquisa; e (3) como parametrizar as estruturas de dados, já que estas parametrizações podem influenciar os resultados obtidos nos diferentes espaços métricos.

Logo, este capítulo está estruturado em três secções. Na primeira secção será descrita a metodologia da avaliação, nomeadamente no que diz respeito aos dados a pesquisar e as parametrizações usadas. De seguida são apresentados os resultados da avaliação para as dimensões timbre e melodia. E por último são apresentadas as conclusões da avaliação da pesquisa por alcance.

### **8.1 Metodologia**

Para uma correcta avaliação das estruturas de dados métricas é necessário ter em atenção a particularidade de que a forma final de certas estruturas depende da ordem de inserção dos elementos do espaço métrico. Com o objectivo de minimizar o impacto deste pormenor nos resultados, de cada coleção de música apresentada anteriormente são criados 3 conjuntos, os quais consistem em permutações aleatórias da respectiva base de dados. Desta forma, as pesquisas realizadas em cada espaço métrico são sempre

efectuadas nestes três conjuntos e os resultados apresentados são a média associada aos valores obtidos nas três permutações.

Em cada permutação associada à base de dados são efectuadas um conjunto de pesquisas por alcance com três raios de pesquisa. Todas as pesquisas por alcance associadas a uma base de dados têm sempre em comum o mesmo conjunto de elementos de pesquisa e os mesmos valores de raios.

Assim sendo, para uma determinada base de dados, um conjunto de pesquisa e um raio, a avaliação da pesquisa por alcance é efectuada da seguinte forma:

- São construídas três instâncias da mesma estrutura de dados métrica, tendo como origem as três permutações da base de dados;
- São realizadas várias pesquisas por alcance nestas estruturas, que correspondem às músicas do conjunto de pesquisa, sendo calculada a média do número de computações de distância que foram efectuadas para realizar as diversas pesquisas, em cada permutação da base de dados;
- O resultado final do número de computações de distância corresponde à média dos valores médios obtidos nas três permutações.

A Figura 51 ilustra o método acima descrito.

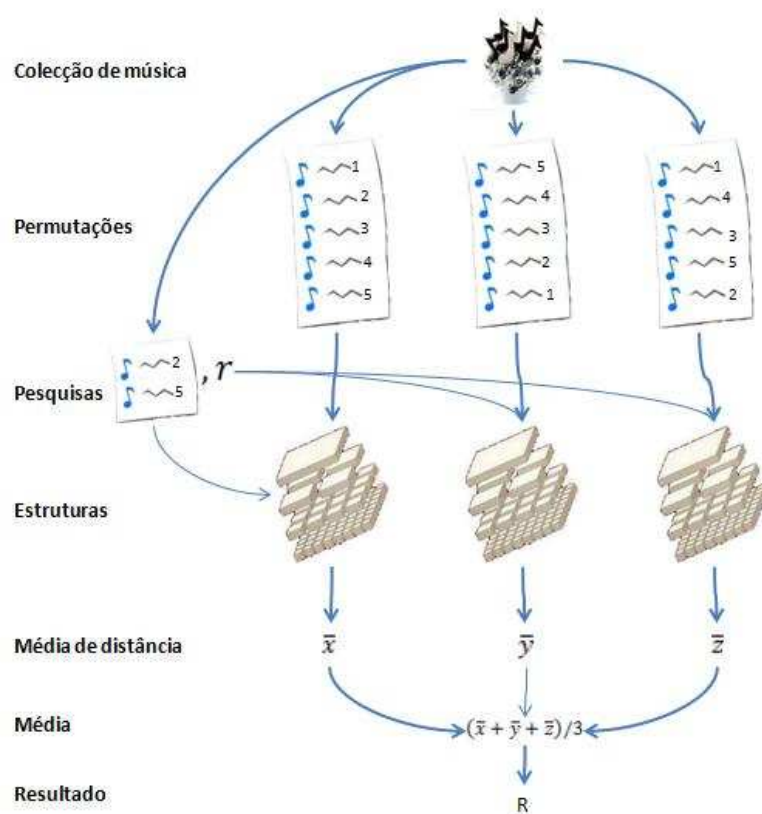


Figura 51. Método de avaliação.

Os resultados experimentais da avaliação final mostram o desempenho das várias estruturas de dados métricas nas pesquisas por alcance sobre os diferentes espaços métricos, tendo como base o número de distâncias calculadas em cada pesquisa.

De seguida serão descritos o conjunto de pesquisa e os raios usados em cada base de dados sobre os espaços métricos. Por último é também descrito as parametrizações usadas nas estruturas de dados métricas.

### **8.1.1 Conjunto de pesquisa**

Para cada base de dados é criado também um conjunto de músicas, representando cerca de 25% da dimensão da respectiva colecção de música, os quais representam os conjuntos de pesquisa de cada base de dados. No conjunto de pesquisa dos espaços métricos da dimensão timbre (secção 7.1.2) foram seleccionados 63 ficheiros que representam 25% de cada artista/álbum. Esta selecção permite obter um conjunto representativo da base de dados, ou seja, todos os artistas, álbuns e géneros musicais estão presentes. Relativamente ao conjunto de pesquisa do espaço métrico da dimensão melodia (secção 7.2.2) foram seleccionados 1103 ficheiros que representam 25% de cada grupo de músicas que possuem o mesmo código na sua designação (ver secção 7.2.2).

### **8.1.1 Raios de pesquisa**

Para realizar a avaliação da eficiência da pesquisa por alcance nas diversas estruturas de dados métricas é necessário definir raios de pesquisa para cada espaço métrico.

Os raios escolhidos para cada espaço correspondem a 40%, 60% e 80% da distância média do espaço métrico. Os raios de pesquisa foram seleccionados com base nas pesquisas efectuadas, onde se verificou experimentalmente, para a base de dados, sobre os espaços métricos da dimensão timbre, que o menor raio seleccionava aproximadamente as músicas do mesmo álbum, o raio intermédio seleccionava as músicas do mesmo artista e algumas de artistas diferentes com géneros musicais muito semelhantes (de acordo com [64]) e por último, o maior raio seleccionava as músicas com géneros musicais semelhantes. Para a base de dados sobre o espaço métrico da dimensão melodia foram usadas as mesmas percentagens para os raios pesquisas, uma vez que esta base de dados apenas contém música folclórica.

Nas tabelas III, IV e V são indicados os valores de cada raio para os diferentes espaços métricos e o número médio de músicas obtidas nas pesquisas por alcance juntamente com a respectiva percentagem do tamanho da colecção de música.

Tabela III. Número médio de músicas e percentagem da colecção de música obtida com a distância de Manhattan.

Distância de Manhattan					
Raio de pesquisa 2,316		Raio de pesquisa 3,5		Raio de pesquisa 4,632	
Número	Percentagem	Número	Percentagem	Número	Percentagem
7,66667	3,07%	38,77778	15,51%	92,09524	36,84%

Tabela IV. Número médio de músicas e percentagem da colecção de música obtida com a distância euclidiana.

Distância euclidiana					
Raio de pesquisa 0,964		Raio de pesquisa 1,5		Raio de pesquisa 1,928	
Número	Percentagem	Número	Percentagem	Número	Percentagem
13,31746	5,33%	59,14286	23,66%	102,95238	41,18%

Tabela V. Número médio de músicas e percentagem da colecção de música obtida com a distância de edição em intervalos melódicos.

Distância de edição em intervalos melódicos					
Raio de pesquisa 21,2		Raio de pesquisa 30		Raio de pesquisa 42,4	
Número	Percentagem	Número	Percentagem	Número	Percentagem
19,69	0,38%	278,34	5,33%	1542,73	29,54%

### 8.1.3 Parametrização das estruturas de dados métricas

No estudo realizado sobre as várias estruturas de dados métricas implementadas é possível observar que estas apresentam parâmetros para a sua construção, com a excepção da estrutura de dados VPTree. Com base nas referências bibliográficas de cada estrutura é observado que a escolha dos valores destes parâmetros afecta de forma significativa a performance das pesquisas por alcance de cada estrutura.

Procurando realizar uma avaliação “justa” da pesquisa por alcance em cada estrutura de dados métrica, foi realizado um estudo, que tem como objectivo obter uma boa parametrização para cada estrutura. Para mais detalhe de como foi efectuado o estudo recomenda-se a leitura do anexo A1, sendo nesta secção apenas indicado os resultados das parametrizações que obtiveram o melhor resultado.

As instâncias das estruturas de dados métricas associadas a cada permutação foram construídas com as parametrizações que serão enumeradas de seguida.



## LAESA

A estrutura de dados métrica LAESA possibilita a parametrização do número de elementos da base de dados que actuam como pivôs. A Tabela VI apresenta o número de pivôs escolhidos nos diferentes espaços métricos.

Tabela VI. Parâmetros da LAESA.

	Intervalos melódicos e distância de edição	Assinatura de MFCCs e distância de Manhattan	Assinatura de MFCCs e distância euclidiana
Número de pivôs	45	16	10

## DSAT, H-DSAT1 e H-DSAT2

A estrutura de dados métrica DSAT e as suas variantes H-DSAT1 e H-DSAT2 possibilitam a parametrização do número de elementos da base de dados em cada nó, ou seja, a aridade da árvore. A Tabela VII apresenta os valores de aridade escolhidos nestas estruturas de dados em cada espaço métrico.

Tabela VII. Parâmetros da DSAT, H-DSAT1 e H-DSAT2.

	Intervalos melódicos e distância de edição	Assinatura de MFCCs e distância de Manhattan	Assinatura de MFCCs e distância euclidiana
DSAT	22	4	2
H-DSAT1	59	5	6
H-DSAT2	59	5	8

## LC

A estrutura de dados métrica LC possibilita duas versões diferentes. Uma versão em que os agrupamentos possuem tamanho fixo e uma outra em que os agrupamentos possuem raio fixo. Na primeira é possível parametrizar o tamanho dos agrupamentos, enquanto na segunda é possível parametrizar o valor do raio dos agrupamentos. A Tabela VIII apresenta a parametrização usada nas duas versões da LC em cada espaço métrico.

Tabela VIII. Parâmetros das duas versões da LC

	Intervalos melódicos e distância de edição	Assinatura de MFCCs e distância de Manhattan	Assinatura de MFCCs e distância euclidiana
LC com agrupamentos de tamanho fixo	2	8	47
LC com agrupamentos de raio fixo	30	7,9	5,88

## GNAT

A estrutura de dados métrica GNAT possibilita a parametrização do número de elementos da base de dados em cada nó que actuam como centros (“degree”). A Tabela IX apresenta os números de centros seleccionados para cada espaço métrico.

Tabela IX. Parâmetros da GNAT.

	Intervalos melódicos e distância de edição	Assinatura de MFCCs e distância de Manhattan	Assinatura de MFCCs e distância euclidiana
Número de centros	100	10	10

## RLC

A estrutura de dados métrica RLC possui dois parâmetros, o raio de construção e a capacidade da RLC (dimensão do “array” de um agrupamento). A Tabela X apresenta os valores do raio de construção e da capacidade da RLC escolhidos em cada espaço métrico.

Tabela X. Parâmetros da RLC.

	Intervalos melódicos e distância de edição	Assinatura de MFCCs e distância de Manhattan	Assinatura de MFCCs e distância euclidiana
Raio de construção	30	6	2,64
Capacidade da RLC	29	38	44

## 8.2 Resultados da avaliação na dimensão timbre

Como, as diferentes estruturas de dados métricas apresentaram um desempenho semelhante nos dois espaços métricos da dimensão timbre, os seus resultados foram descritos numa única secção.

A Figura 52 ilustra os resultados das estruturas de dados métricas nos dois espaços métricos da dimensão timbre com os três raios de pesquisa, isto é, o número médio de distâncias calculadas em cada pesquisa.

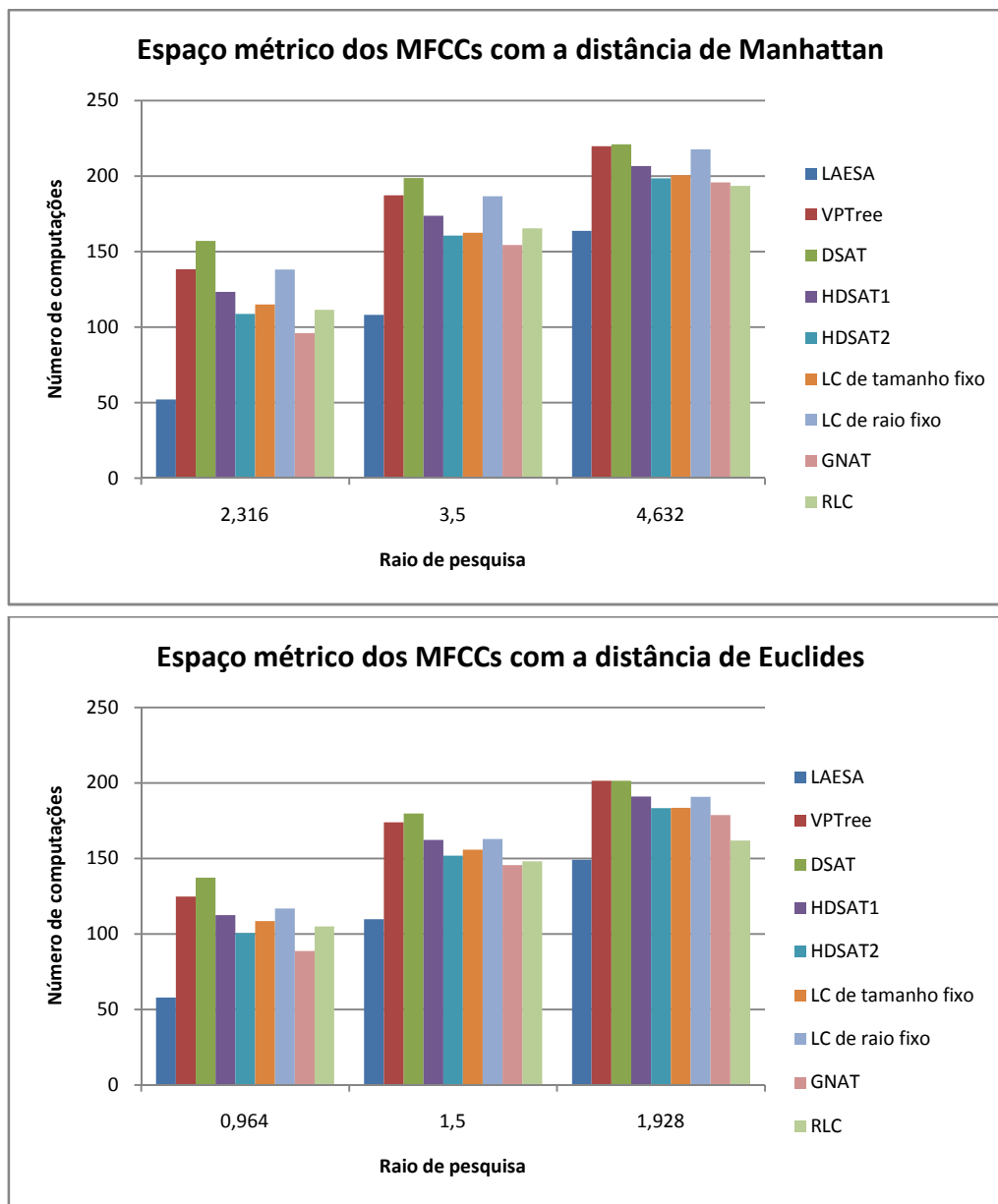


Figura 52. Número de distâncias calculadas em cada espaço métrico da dimensão timbre.

A Tabela XI sumariza os resultados apresentados nos gráficos da Figura 52 e permite observar as percentagens do número de computações de distâncias em relação ao tamanho da base de dados realizadas em cada espaço métrico nas respectivas pesquisas por alcance.

Tabela XI. Percentagem dos ficheiros comparados da base de dados da dimensão timbre nas várias pesquisas por alcance.

	MFCCs e distância de Manhattan			MFCCs e distância euclidiana		
Raio de pesquisa	2,316	3,5	4,632	0,964	1,5	1,928
LAESA	21%	43%	66%	23%	44%	60%
VPTree	55%	75%	88%	50%	70%	81%
DSAT	63%	79%	88%	55%	72%	81%
HDSAT1	49%	70%	83%	45%	65%	76%
HDSAT2	43%	64%	79%	40%	61%	73%
LC de tamanho fixo	46%	65%	80%	43%	62%	73%
LC de raio fixo	55%	75%	87%	47%	65%	76%
GNAT	38%	62%	78%	36%	58%	71%
RLC	45%	66%	77%	42%	59%	65%

Com base nos resultados dos gráficos da Figura 53 e da tabela XI, é possível observar que a LAESA nestes dois espaços métricos obteve os melhores resultados em todas as pesquisas (cor verde na Tabela XI). Pondo de parte a estrutura LAESA, as restantes estruturas demonstraram um comportamento bastante competitivo (resultados próximos), sendo as estruturas GNAT, H-DSAT2 e RLC as que melhores resultados apresentaram. No entanto é de realçar que a RLC se desmarca na pesquisa com maior raio, em ambos os espaços métricos, apresentando melhores resultados que as restantes.

As estruturas que apresentaram os piores resultados nestes espaços métricos foram a DSAT e a VPTree (cor vermelha na Tabela XI), onde a DSAT apresentou sempre os piores resultados em todas as pesquisas.

Relativamente aos dois grupos de estruturas, o grupo das LCs e o grupo da DSAT e suas variantes HDSAT1 e HDSAT2, é possível observar que:

- Do grupo das LCs, a LC de tamanho fixo registou sempre melhor performance, ainda que ténue;
- No grupo das DSATs manteve-se sempre a mesma ordem de performance nas estruturas de dados com os três raios de pesquisa. Sendo a DSAT a estrutura com pior performance e a H-DSAT2 com melhor performance;

Em relação às estruturas de dados dinâmicas é verificado que a H-DSAT2 e a RLC (cor azul e verde seco, respectivamente) apresentam resultados muito semelhantes (na maioria das situações com uma diferença de 2%), no entanto no maior raio com a distância euclidiana, a RLC obtém um resultado claramente superior.

Por último, é possível observar que à medida que o raio de pesquisa aumenta, a LAESA apresenta a maior degradação e a DSAT apresenta a menor degradação, embora esta última seja a estrutura que pior performance apresentou. Relativamente às estruturas dinâmicas H-DSAT2 e RLC apresentaram uma degradação ligeira, sendo a RLC a estrutura que menor degradação apresentou de um modo geral.

### 8.3 Resultados da avaliação na dimensão melodia

A Figura 53 ilustra os resultados das estruturas de dados métricas no espaço métrico da dimensão melodia com os três raios de pesquisa.

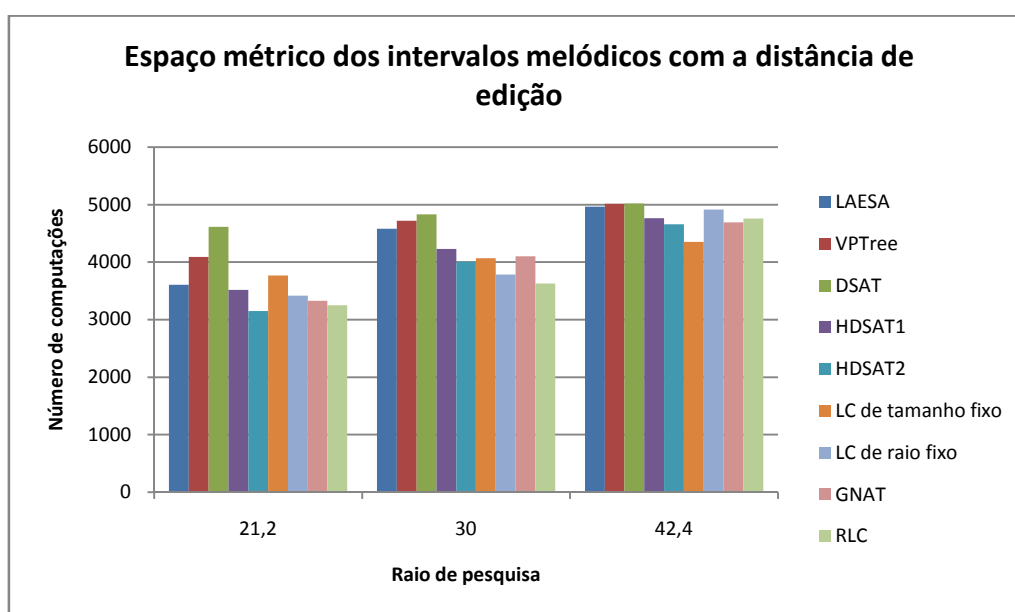


Figura 53. Número de distâncias calculadas no espaço métrico da dimensão melodia.

A Tabela XII sumariza os resultados apresentados nos gráficos da Figura 53 e permite observar as percentagens do número de computações de distâncias em relação ao tamanho da base de dados realizadas nas várias pesquisas por alcance no espaço métrico da dimensão melodia.

Tabela XII. Percentagem dos ficheiros comparados da base de dados da dimensão melodia nas várias pesquisas por alcance.

	Intervalos melódicos e distância de edição		
Raio de pesquisa	21,2	30	42,4
LAESA	69%	88%	95%
VPTree	78%	90%	96%
DSAT	88%	93%	96%
HDSAT1	67%	81%	91%
HDSAT2	60%	77%	89%
LC de tamanho fixo	72%	78%	83%
LC de raio fixo	65%	72%	94%
GNAT	64%	79%	90%
RLC	62%	70%	91%

Com base nos resultados dos gráficos da Figura 53 e da Tabela XII, é possível observar que em cada raio de pesquisa houve diferentes estruturas que obtiveram o melhor resultado (cor verde na Tabela XII). No menor raio a H-DSAT2 obteve o melhor resultado, no raio do meio a RLC obteve o melhor resultado e no maior raio a LC de tamanho fixo obteve o melhor resultado. Pondo de parte a estrutura DSAT na pesquisa com menor raio, as várias estruturas revelaram um comportamento bastante competitivo (resultados próximos), sendo as estruturas H-DSAT2 e RLC as que melhores resultados apresentaram, embora a RLC se tenha desmarcado na pesquisa com raio intermédio.

As estruturas que apresentaram os piores resultados nestes espaços métricos foram a DSAT e a VPTree (cor vermelha na Tabela XII), onde a DSAT apresentou sempre os piores resultados em todas as pesquisas.

Relativamente aos dois grupos de estruturas, o grupo das LCs e o grupo da DSAT e suas variantes HDSAT1 e HDSAT2, é possível observar que:

- do grupo das LCs, a LC de raio fixo registou melhor performance no raio menor e no raio intermédio, enquanto a LC de tamanho fixo apresentou um desempenho superior no raio maior (aproximadamente 9%).
- Do grupo das DSATs manteve-se sempre a mesma ordem de performance nos três raios de pesquisa. Sendo a DSAT a estrutura com pior performance e a H-DSAT2 com melhor performance.

Em relação às estruturas de dados dinâmicas é verificado que a H-DSAT2 e a RLC (cor azul e verde seco, respectivamente) apresentam resultados muito semelhantes

(na maioria das situações uma diferença de 2%), no entanto no raio intermédio a RLC obtém um resultado claramente superior (aproximadamente 7%).

Por último, é possível observar que à medida que o raio de pesquisa aumenta, a DSAT apresenta a menor degradação, embora esta seja a estrutura que pior performance apresentou, enquanto a LC (de tamanho fixo) apresenta uma degradação semelhante mas com melhor performance. Relativamente às estruturas dinâmicas H-DSAT2 e RLC, verificou-se uma degradação significativa de um modo geral, sendo a RLC a estrutura que apresentou uma fraca degradação entre os dois primeiros raios e uma elevada degradação entre os últimos dois.

## **8.4 Conclusão**

Para pesquisas exaustivas, é necessário para cada situação comparar a música com a totalidade dos documentos musicais presentes na base de dados. Com o estudo realizado neste capítulo é possível observar que todas as estruturas de dados métricas possibilitaram a redução do número de comparações, em cada espaço métrico. Nos espaços métricos da dimensão timbre registou-se uma redução significativa do número de comparações, num intervalo de 80% com o menor raio de pesquisa a 20% com o maior raio de pesquisa. Enquanto no espaço métrico da melodia apenas se verificou uma ligeira redução do número de computações, aproximadamente um intervalo de 42% com o menor raio de pesquisa a 5% com o maior raio de pesquisa. Com base nisto, é possível concluir que a dimensão timbre apresentou resultados satisfatórios, embora aquém dos esperados nos raios maiores. Relativamente à dimensão melodia, esta apresentou maus resultados face à perspectiva que se tinha.

Todas as estruturas de dados obtiveram piores resultados (em percentagem) no espaço métrico da dimensão melodia com intervalos melódicos e a distância de edição.

Com base nos resultados obtidos na avaliação da pesquisa por alcance nos três espaços métricos concluí-se que do grupo das DSATs, se manteve sempre a mesma ordem de performance nos três raios de pesquisa. Sendo a DSAT a estrutura com pior performance e a H-DSAT2 com melhor performance. Seguramente esta ordem é justificada pelo facto das variantes da DSAT utilizarem elementos pivôs, sendo a H-DSAT2 a que utiliza mais pivôs. No entanto, no que diz respeito à degradação destas estruturas verificou-se a situação inversa, onde a DSAT apresenta a menor degradação e a H-Dsat2 a maior degradação.

No que diz respeito às estruturas dinâmicas H-DSAT2 e RLC, estas apresentaram um comportamento bastante competitivo nos três espaços métricos.

Por último, é possível observar, que a estrutura DSAT apresenta os piores resultados e também a menor degradação nos três espaços métricos estudados.



## 9. Protótipo

Neste capítulo é descrito o protótipo final realizado durante a fase de elaboração desta dissertação.

O protótipo foi implementado na linguagem JAVA e reúne a implementação dos espaços métricos e do pacote de estruturas de dados métricas anteriormente descritos nos capítulos 7 e 6, respectivamente. Este protótipo permite a geração de “playlists” de músicas semelhantes utilizando pesquisas por alcance em estruturas de dados métricas. Dada uma música e um raio de pesquisa, o sistema devolve um conjunto de músicas semelhantes do espaço métrico seleccionado, de forma a obter uma “playlist”.

Este capítulo está estruturado em quatro secções. A primeira apresenta a interface do protótipo, a segunda a sua arquitectura e a terceira apresenta detalhes do processamento dos dados. A última secção apresenta as conclusões.

### 9.1 Interface

A Figura 58 ilustra a interface apresentada pelo sistema de geração de “playlists”. São visíveis várias componentes de parametrização e de visualização:

- M1 – “Metric Space”: Este menu permite a selecção do espaço métrico no qual se pretende gerar uma “playlist”. Quando é seleccionado um espaço métrico, o menu M2 é inicializado com uma música da base de dados utilizada nesse espaço métrico;
- M2 – “Song Query”: Este menu permite seleccionar a música de pesquisa, com a qual se pretende gerar uma “playlist” de músicas semelhantes à música seleccionada. Na alteração da música de pesquisa, o leitor L1 é inicializado com a nova música seleccionada;
- M3 – “Data Structure”: Este menu permite seleccionar a estrutura de dados métrica com a qual se pretende efectuar a pesquisa por alcance;
- M4 – “Search Radius”: Este menu permite seleccionar o raio de pesquisa;

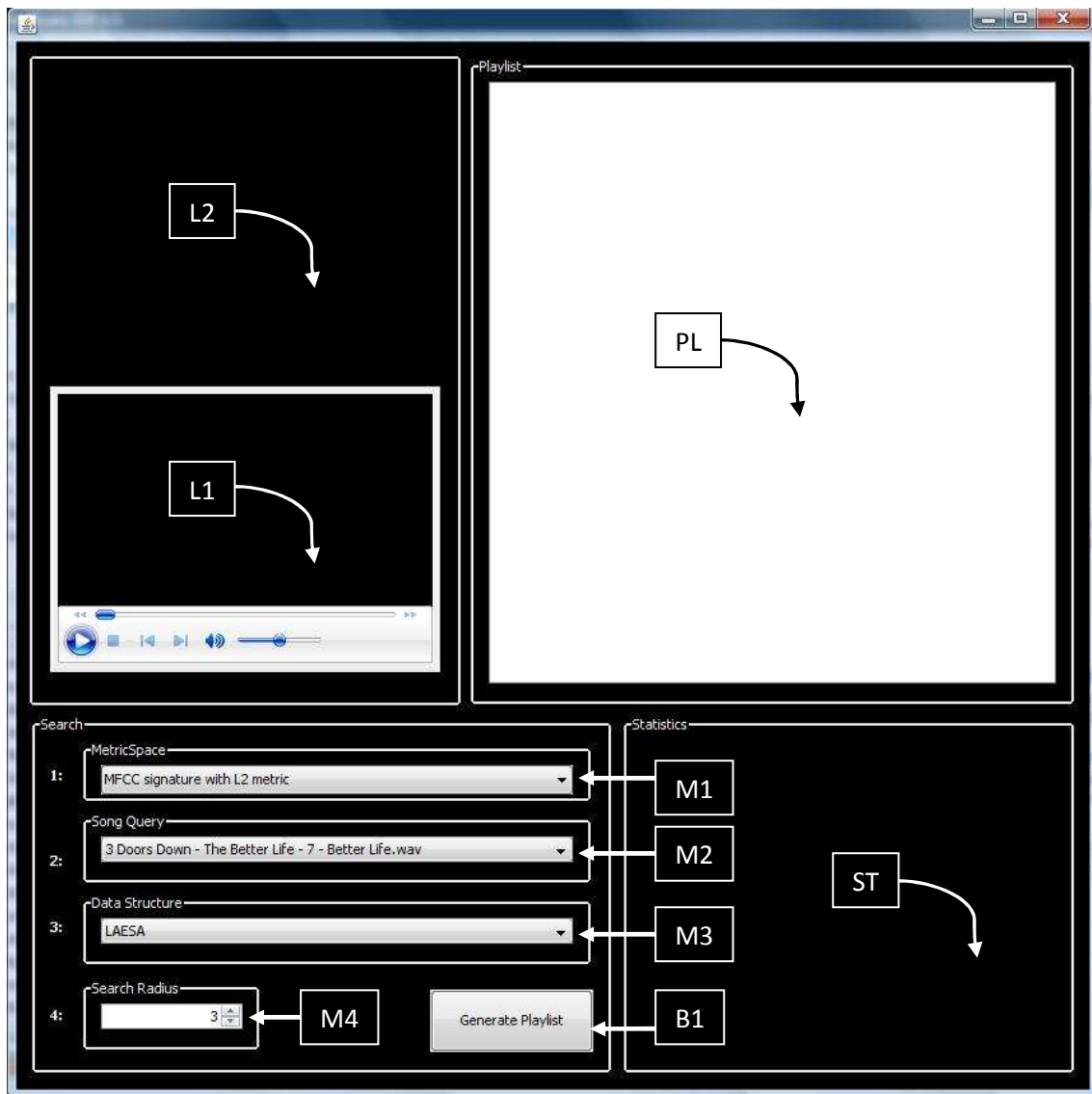


Figura 54. Interface do sistema.

- B1 – “Generate Playlist”: Este botão permite gerar uma “playlist” através de uma pesquisa por alcance com base nos parâmetros definidos nos menus: M1, M2, M3 e M4. A componente PL é actualizada com o conjunto de músicas da “playlist”, obtida pela pesquisa por alcance com os parâmetros seleccionados. A componente ST é actualizada com estatísticas referentes à pesquisa efectuada;
- ST – “Statistics”: Esta componente permite a exibição de estatísticas referentes à pesquisa efectuada. Estas estatísticas são: número de músicas da “playlist”, tamanho da “playlist” em percentagem do número de músicas do espaço métrico, número de cálculos de distâncias efectuados e por último,

número de cálculos de distâncias efectuados em percentagem do número de músicas do espaço métrico;

- PL – “Playlist”: Permite a exibição da lista de músicas da “playlist” gerada pela pesquisa por alcance. Na selecção de uma música é exibido o leitor L2 inicializado com a música seleccionada;
- L1: Leitor de música que permite a reprodução da música de pesquisa;
- L2: Leitor de música que permite a reprodução da música seleccionada na “playlist” (componente PL).

A Figura 55 ilustra a interface do sistema após a geração de uma “playlist” para a música de pesquisa “Disturbed - The Sickness - 04 - Down With The Sickness.wav” no espaço métrico da distância euclidiana na dimensão timbre. No exemplo apresentado foi escolhida a estrutura RLC com um raio de pesquisa igual a 1,2.

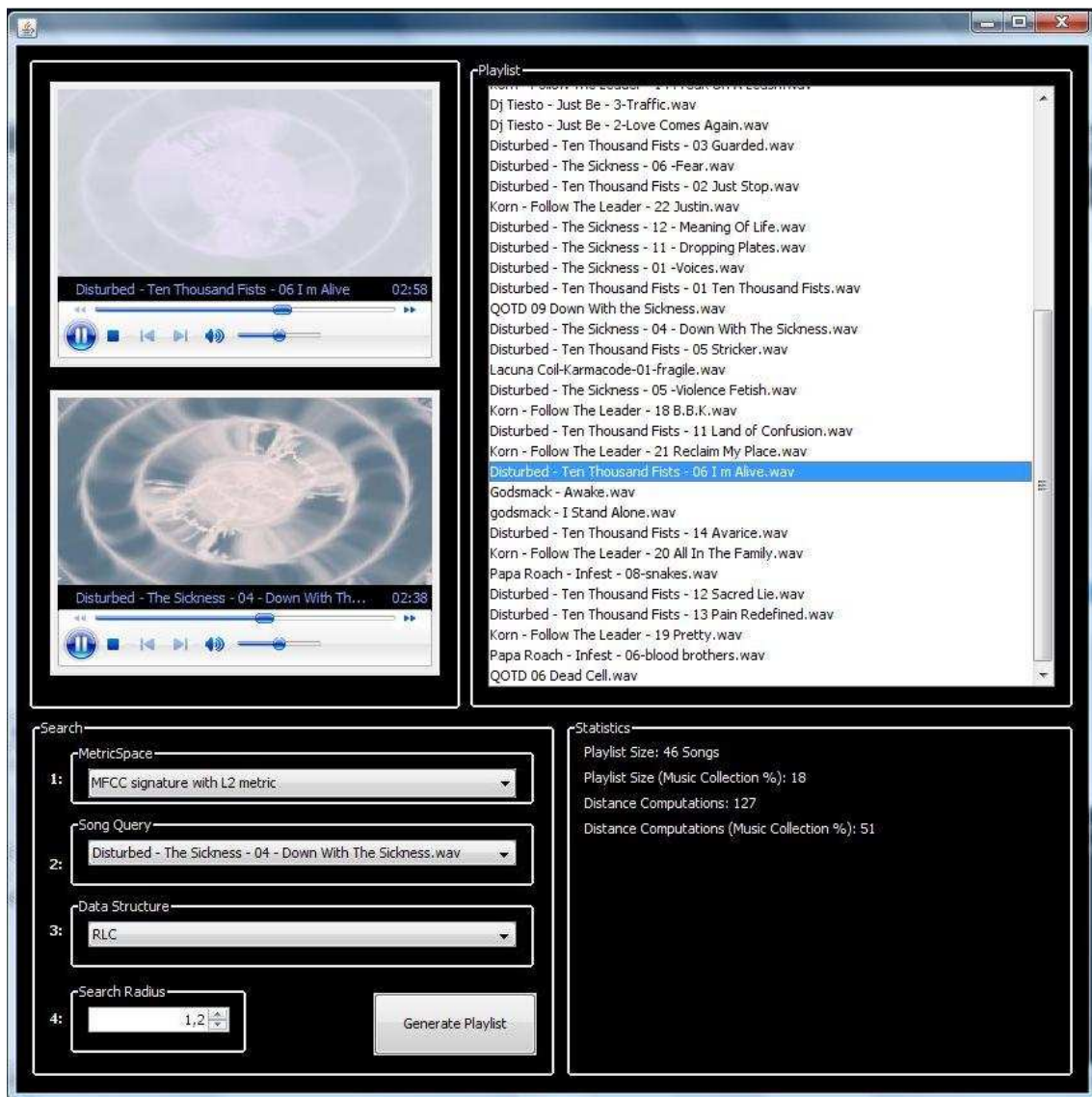


Figura 55. Exemplo da interface do sistema após a geração de uma “playlist”.

## 9.2 Arquitectura

A Figura 56 ilustra a arquitectura do sistema, nomeadamente o diagrama de pacotes.

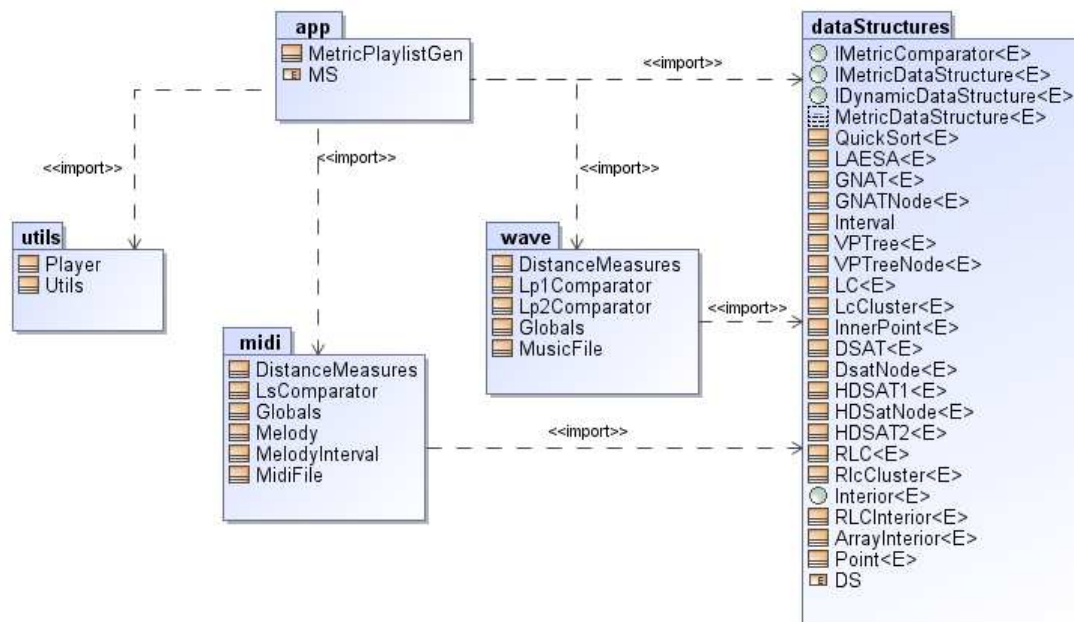


Figura 56. Diagrama de pacotes do Protótipo.

Os pacotes “midi” e “wave” contêm a implementação dos espaços métricos da dimensão melodia e timbre, respectivamente. Os espaços métricos utilizados neste protótipo encontram-se armazenados na directoria “/MetricSpaces”, onde para cada espaço existe uma directoria com o respectivo nome e que contém os seus elementos. Estes elementos, de um espaço métrico, consistem na sua colecção de música e num ficheiro “dbs” que contém uma representação da colecção de música já processada pelo sistema, ou seja, cada música já possui os seus descritores. Este último ficheiro não é obrigatoriamente necessário para a execução do sistema, sendo apenas gerado (aquando da sua omissão) como forma de aumentar a performance, no momento de carregar o espaço métrico.

O pacote “dataStructures” corresponde ao pacote de estruturas de dados métricas descrito anteriormente no capítulo 6. Neste protótipo foram utilizadas todas as estruturas de dados métricas que fazem parte do pacote: LAESA, VPTree, DSAT, HDSAT1, H-DSAT2, LC com raio fixo, LC com tamanho fixo, GNAT, e por último, a RLC. As estruturas de dados métricas foram parametrizadas com as configurações descritas na secção 8.2.

O pacote “utils” contém funções utilitárias e a implementação do leitor de música utilizado no protótipo.

Por último, o pacote “app” contém a implementação da interface do protótipo e a classe que contém o fluxo de controlo do sistema.

### 9.3 Processamento

No arranque do sistema, os menus M1, M2, M3 e M4 são definidos por defeito com os seguintes parâmetros:

- M1: espaço métrico da dimensão timbre com a distância euclidiana;
- M2: com uma música da base de dados do espaço métrico definido em M1;
- M3: com a estrutura de dados métrica LAESA;
- M4: com o raio de pesquisa por defeito igual a 3;

Durante a execução do programa existem situações em que é realizado processamento de dados, estas situações estão relacionadas principalmente com as alterações dos parâmetros dos menus M1, M2, M3, M4 e com o botão “Generate Playlist”.

No arranque do sistema e quando é seleccionado um novo espaço métrico através do menu M1, é necessário carregar os elementos do novo espaço em memória. Deste modo, é verificado se o espaço métrico já possui o seu ficheiro “dbs”. Em caso afirmativo, a base de dados de música juntamente com os descritores de cada música é carregada. Caso contrário, é necessário percorrer a colecção de música do espaço métrico, criar instâncias dos objectos que representam cada música e calcular os seus descritores, após este processo é escrito em disco o ficheiro “dbs” relativo à base de dados do espaço métrico. No fim do carregamento dos elementos do espaço métrico são criadas as várias estruturas de dados métricas instanciadas com os elementos do espaço métrico. Esta decisão de implementar a criação de todas as estruturas de dados métricas neste momento baseia-se nos pressupostos que um utilizador quando selecciona um espaço métrico pretende gerar várias “playlists” com estruturas diferentes e que a alteração do espaço métrico consiste numa operação efectuada com baixa frequência.

Quando é seleccionada uma música no menu M2, o leitor L1 é inicializado com esta música, de modo a permitir a sua reprodução.

A selecção de uma estrutura de dados métrica no M3 não causa qualquer processamento uma vez que todas as estruturas foram instanciadas no momento de carregamento do espaço métrico, sendo apenas guardada a indicação da estrutura seleccionada.

A parametrização do raio de pesquisa no menu M4, à semelhança do que sucede no menu M3, também não produz nenhum processamento, sendo apenas guardado o valor definido.

Por último, a execução do botão B1 inicia a pesquisa por alcance na estrutura de dados métrica seleccionada com a música e raio de pesquisa definidos em M2 e M4, respectivamente. Esta pesquisa produz como resultado um conjunto de músicas que é exibido na componente PL da interface.

#### **9.4 Conclusões**

Neste capítulo foi descrito um protótipo que permite testar o desempenho das várias estruturas de dados métricas na pesquisa por alcance no domínio das aplicações de semelhança em música.

O protótipo permite gerar “playlists” de música semelhante com base no método de pesquisa por alcance em diferentes estruturas de dados métricas, apresentando as estatísticas relativas à geração da “palylist”. Com base na interface que o protótipo disponibiliza é possível observar que este exhibe o conjunto de músicas da “playlist” gerada e permite também a reprodução das músicas contidas na “playlist”.

## 10. Conclusões e trabalho futuro

Neste capítulo são apresentadas as conclusões e contribuições finais relativas a este projecto de investigação, bem como as possíveis direcções de investigação futura neste domínio de aplicação.

### 10.1 Conclusões

Com o objectivo de estudar a aplicabilidade e eficiência do uso de estruturas de dados métricas para a geração de “playlists” de músicas semelhantes, nesta dissertação elaboraram-se dois estudos: o estudo da aplicabilidade e o estudo da eficiência. De forma a estudar a aplicabilidade das estruturas de dados métricas em aplicações de semelhança de música, foram realizadas as seguintes actividades nesta dissertação:

- Descrição das várias técnicas para a representação de conteúdo musical em três dimensões da música: melodia, timbre e ritmo;
- Descrição das diversas medidas de semelhança do conteúdo musical;
- Implementação de três espaços métricos nas dimensões melodia e timbre;
- Validação “informal” destes espaços métricos, isto é, se estes exprimem a similaridade de músicas correctamente.

Com base nas tarefas efectuadas acima, conclui-se que é possível definir espaços métricos que exprimem satisfatoriamente a similaridade de músicas e que, por conseguinte, as estruturas de dados métricas são aplicáveis neste domínio de aplicação. É de salientar que a avaliação dos espaços métricos consistiu um processo muito complicado e subjectivo.

Quanto ao estudo da eficiência das estruturas de dados métricas em aplicações de semelhança de música, foram realizadas as seguintes actividades:

- Descrição de nove estruturas de dados métricas;
- Implementação de um pacote genérico que inclui nove estruturas de dados métricas;

- Avaliação da eficiência da pesquisa por alcance das estruturas de dados em duas bases de dados sobre três espaços métricos. Salienta-se aqui que cada base de dados foi usada em espaços métricos diferentes.

Com base na avaliação realizada é possível concluir que o uso das estruturas de dados métricas representa uma melhoria no processo de geração de “playlists”, uma vez que, se verificou sempre uma redução dos cálculos de semelhança face ao tamanho da colecção de música. Tendo como base o desenvolvimento deste tipo de aplicação, faz mais sentido o uso de estruturas de dados métricas dinâmicas. Por último, é de salientar que as estruturas de dados métricas dinâmicas RLC e H-DSAT2 foram muito competitivas.

## 10.2 Contribuições

Durante a realização desta dissertação foram alcançadas todas as contribuições previstas no início desta dissertação (secção 1.3):

- Um estudo de vários espaços métricos existentes, no qual são descritas várias possíveis representações e funções de semelhança de música;
- Uma avaliação da eficiência da pesquisa por alcance nas nove estruturas de dados métricas nos vários espaços métricos implementados sobre as respectivas bases de dados (colecções de música) em duas dimensões: melodia e timbre.

Para além destas contribuições previamente estabelecidas, é importante realçar que o estudo realizado permitiu ainda oferecer à comunidade científica:

- Protótipos que listam as músicas da base de dados por ordem decrescente de semelhança à música de pesquisa seleccionada de acordo com a medida de distância. Estes protótipos podem ser utilizados em futuros projectos para uma avaliação dos espaços métricos;
- Protótipo que permite a geração de “playlists” de música semelhante baseado no conteúdo musical por aplicação de pesquisas por alcance com estruturas de dados métricas. Este protótipo poderá ser utilizado posteriormente em futuros estudos;
- Pacote genérico de estruturas de dados métricas implementado em JAVA que pode ser reutilizado em diferentes domínios, permitindo um rápido



desenvolvimento de futuros estudos que envolvam estruturas de dados métricas;

- Um artigo científico [75] que serviu de teste para uma avaliação a nível internacional, no qual se avalia as estruturas de dados métricas em espaços métricos, na dimensão timbre.

### **10.3 Trabalho futuro**

Relativamente aos temas que possam ser abordados futuramente de modo a melhorar o estudo realizado, seria interessante:

- Implementar um espaço métrico na dimensão ritmo;
- Realizar um inquérito para avaliação da medida de similaridade dos diversos espaços métricos;
- Implementar e avaliar a eficiência da pesquisa dos  $k$  vizinhos mais próximos nas diferentes estruturas de dados métricas;
- Analisar a complexidade espacial das várias estruturas de dados métricas, permitindo assim obter um novo critério para observar a eficiência das estruturas de dados;
- Acrescentar as funcionalidades de adicionar e remover músicas dos espaços métricos no protótipo final;
- Avaliar a degradação do desempenho das estruturas de dados métricas dinâmicas provocada pelas operações de inserção e remoção;
- Modificar o protótipo final de modo a torná-lo genérico, de forma a permitir a adição de novos espaços métricos.



## 11. Bibliografia

- [1] M. Skalak, J. Han, B. Pardo, “Speeding Melody Search With Vantage Point Trees”, *Proc. of ISMIR* 2008.
- [2] B. Pardo, W. Birmingham, “Encoding Timing Information for Musical Query Matching”, *Proc. of ISMIR* 2002.
- [3] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, G. Tzanetakis, “A Comparative Evaluation of Search Techniques for Query-by-Humming Using the MUSART Testbed”, *Journal of the American Society for Information Science and Technology*, 2007: p. 687 – 701.
- [4] M. Ryynänen, A. Klapuri, “Query by humming of Midi and audio using locality sensitive hashing”, *Proc. of ICASSP* 2008.
- [5] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, R. van Oostrum, “Using Transportation Distances for Measuring Melodic Similarity”, *Proc. of ISMIR* 2003.
- [6] Y. Rubner, C. Tomasi, L. Guibas, “The Earth Mover's Distance as a Metric for Image Retrieval”, *Tech. Rep.*, Stanford University, 1998.
- [7] Y. Rubner, Source code for the Earth Mover's Distance software, 1998, disponível em <http://robotics.stanford.edu/~rubner/emd/default.htm>.
- [8] P. P. Giannopoulos, R. C. Veltkamp, “A Pseudo-Metric for Weighted Point Sets”, *Proc. of the 7th European Conference on Computer Vision (ECCV)*, pages 715–730, Copenhagen, Denmark, 2002 Springer-Verlag.
- [9] Rainer Typke, “*Music Retrieval based on Melodic Similarity*”, Doctoral Thesis, Universiteit Utrecht 2007.
- [10] R. Typke, F. Wiering, R. C. Veltkamp, “Evaluating the Earth Mover's Distance for measuring symbolic melodic similarity”, disponível em <http://www.music->

ir.org/evaluation/mirex-results/articles/similarity/typke.pdf consultado em Janeiro de 2009.

[11] R. Typke, F. Wiering, R. C. Veltkamp, “MIREX symbolic melodic similarity and query by singing/humming”, disponível em [http://www.music-ir.org/evaluation/MIREX/2006\\_abstracts/SMS\\_QBSH\\_typke.pdf](http://www.music-ir.org/evaluation/MIREX/2006_abstracts/SMS_QBSH_typke.pdf) consultado em Janeiro de 2009.

[12] S. Salvador, P. Chan, “FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space”, *Intelligent Data Analysis, Volume 11, Issue 5* (October 2007) Pages 561-580.

[13] L. Chen, “*Similarity search over time series and trajectory data*”, Doctoral Thesis, University of Waterloo, Ontario, Canada, 2005.

[14] D. Mazzoni, R. B. Dannenberg, “Melody Matching Directly From Audio”, *Proc. of ISMIR* 2001.

[15] N. Orio, “Music Retrieval: A Tutorial and Review”, *Foundations and Trends in Information Retrieval Volume 1, No1* (November 2006) Pages: 1 – 90.

[16] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling”, *Proc. of ISMIR*, 2000.

[17] J. T. Foote, “Content-based retrieval of music and audio”, in *SPIE*, 1997, pp. 138–147.

[18] B. Logan, A. Salomon, “A music similarity function based on signal analysis”, *Proc. of International Conference on Multimedia and Expo (ICME)*, August 2001.

[19] J.-J. Aucouturier, F. Pachet, “Music Similarity Measures: What’s the Use?”, *In Proc ISMIR* 2002.

[20] E. Pampalk, “A Matlab toolbox to compute music similarity from audio”, *In Proc ISMIR* 2004.

[21] M. Mandel, D. Ellis, “Song-level features and support vector machines for music classification”, *In Proc ISMIR* 2005.

- [22] G. Tzanetakis, P. Cook, “Musical Genre Classification of Audio Signals”, *Proc. of IEEE Transactions on Speech and Audio Processing*, 2002.
- [23] K. Jacobson, “A Multifaceted Approach to Music Similarity”, *In Proc ISMIR* 2006.
- [24] J. Foote, M. Cooper, U. Nam, “Audio Retrieval by Rhythmic Similarity.” *In Proc ISMIR* 2002.
- [25] J. Foote, U. Nam “The Beat Spectrum: A New Approach to Rhythm Analysis,” *In Proc. ICME* 2001.
- [26] C. Schmidt-Jones, “Timbre: The Color of Music”, disponível em <http://cnx.org/content/m11059/latest/> consultado em Janeiro de 2009.
- [27] L. R. Rabiner, B. H. Juang, “*Fundamentals of Speech Recognition*”, Prentice–Hall, 1993.
- [28] W. Gangbo, “An Introduction to the Mass Transportation Theory and its Applications”, School of Mathematics, Georgia Institute of Technology, 2004.
- [29] S. Eklisheva, C. Houdré, “Transportation Distance and the Central Limit Theorem”, disponível <http://www.math.gatech.edu/~houdre/research/papers/TransportCLT.pdf> consultado em Janeiro de 2009.
- [30] GUIDO Music Notation, “The GUIDO NoteServer”, disponível em <http://www.noteserver.org> consultado em Janeiro de 2009.
- [31] MuseData, “An electronic library of classical music scores”, disponível em <http://www.musedata.org> consultado em Janeiro de 2009.
- [32] MusicXML, “Recordare: Internet music publishing and software”, disponível em <http://www.musicxml.org> consultado em Janeiro de 2009.
- [33] LilyPond, “LilyPond ... music notation for everyone”, disponível em <http://lilypond.org/web/> consultado em Janeiro de 2009.
- [34] MIDI Manufacturers Association, “Summary of MIDI Messages”, <http://www.midi.org/about-midi/table1.shtml> consultado em Janeiro de 2009.

- [35] K. Brandenburg, “MP3 and AAC explained”, *In Proc AES 17th Inter-national Conference on High-Quality Audio Coding*, 1999.
- [36] G. Tzanetakis, P. Cook, “Sound analysis using MPEG compressed audio”, *In Proc ICASSP* 2000.
- [37] Chih-Chin Liu, Po-Jun Tsai, “Content-based retrieval of MP3 music objects” *In Proc of the tenth international conference on Information and knowledge management*.
- [38] J. M. Martínez, “MPEG-7 Overview,” (version 10), disponível em <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm> consultado em Janeiro de 2009.
- [39] Hyoung-Gook Kim, T. Sikora, “Comparison of MPEG-7 Audio Spectrum Projection Features and MFCC applied to Speaker Recognition, Sound Classification, and Audio Segmentation”, *Proc. of ICASSP* 2004.
- [40] B. Logan, “Content-Based Playlist Generation: Exploratory Experiments”, *In Proc ISMIR* 2002
- [41] E. Chávez, G. Navarro, “A compact space decomposition for effective metric indexing”, *ACM Computing Surveys*, 33(3):273-321, 2001.
- [42] Margarida Mamede, “Recursive List of Clusters: A Dynamic Data Structure for Range Queries in Metric Spaces”, *Proc. of 20<sup>th</sup> International Symposium on Computer and Information Sciences (ISCIS 2005)*, *Lectures Notes in Computer Science*, vol 3733. Berlin, Germany. Springer-Verlag, 2005, pages 843-853.
- [43] P. N. Yianilos, “Data structures and algorithms for nearest neighbor search in general metric spaces”, *Proc. of the 4th Annual ACMSIAM Symposium on Discrete Algorithms (SODA’93)*, pages 311–321, Philadelphia, PA, 1993.
- [44] P. Zezula, G. Amato, V. Dohnal, M. Batko, “Similarity Search The Metric Space Approach”, *Series: Advances in Database Systems* , Vol. 32, Springer 2006.
- [45] S. Brin, “Near neighbor search in large metric spaces”, *Proc. of the 21st International Conference on Very Large Data Bases (VLDB’95)*, pages 574–584, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

- [46] Wikipedia, “Voronoi diagram”, disponível em [http://en.wikipedia.org/wiki/Dirichlet\\_domain](http://en.wikipedia.org/wiki/Dirichlet_domain), consultado em Janeiro de 2009
- [47] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín, “Searching in metric spaces”, *ACM Computing Surveys*, 33(3):273–321, 2001.
- [48] D. Arroyuelo, G. Navarro, N. Reyes, “Fully Dynamic and Memory-Adaptive Spatial Approximation Trees”, *In Proc. CACIC'03*, pages 1502-1513.
- [49] D. Arroyuelo, F. Muñoz, G. Navarro e N. Reyes, “Memory-Adaptive Dynamic Spatial Approximation Trees” *Proc. of the 10th International Symposium on String Processing and Information Retrieval*, pages 360-368, 2003.
- [50] G. Navarro, N. Reyes, “Fully Dynamic Spatial Approximation Trees”, *Proc. of the 9th International Symposium on String Processing and Information Retrieval (SPIRE 2002)*, volume 2476 of Lecture Notes in Computer Science, pages 254–270, Berlin, Germany, 2002. Springer-Verlag.
- [51] M. L. Micó, J. Oncina, E. Vidal, “A new version of the nearest-neighbour approximating and eliminating search algorithm (AESAs) with linear preprocessing time and memory requirements”, *Pattern Recognition Letters*, 15(1):9–17, 1994.
- [52] Ismir, “International Society for Music Information Retrieval”, disponível em <http://www.ismir.net/> consultado em Janeiro de 2009.
- [53] L. Micó, J. Oncina, E. Vidal, “An Algorithm for Finding Nearest Neighbours in Constant Average Time With a Linear Space Complexity”, *Pattern Recognition*, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, *Proc. of 11<sup>th</sup> IAPR International Conference*.
- [54] K. West, P. Lamere, “A Model-Based Approach to Constructing Music Similarity Functions”, *EURASIP Journal on Advances in Signal Processing*, Pages: 149 – 149, vol. 2007.
- [55] C. McKay, “Automatic Music Classification and Similarity Analysis”, Course Paper Université de Montreal, 2005.

- [56] MPEG, “International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11, Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio”, Motion Pictures Expert Group, 1992.
- [57] M. Mamede, F. Barbosa, “Range Queries in Natural Language Dictionaries with Recursive List of Clusters”, *Proc. of 22<sup>nd</sup> International Symposium on Computer and Information Sciences (ISCIS 2007)*, Ankara, Turkey, IEEExplore, 2007, doi: 10.11.09/ISCIS.2007.4456857.
- [58] F. Barbosa, “Similarity-based retrieval in high-dimensional data with Recursive Lists of Clusters: A study case with Natural Language Dictionaries”, to appear on *International Conference on Information management and engineering (ICIME 2009)*, Kuala Lumpur, Malaysia.
- [59] Midi, “Musical Instrument Digital Interface”, disponível em <http://en.wikipedia.org/wiki/Midi>, consultado em Janeiro de 2009.
- [60] M. Mamede, “A Dynamic Data Structure for Range Queries in High Dimensional Metric Spaces”, disponível em <http://www-ctp.di.fct.unl.pt/~mm/home/dynamic-07.pdf>, consultado em Janeiro de 2009.
- [61] Alfie Tan Kok Leong, “A Music Identification System Based on Audio Content Similarity”, Doctoral Thesis, University of Queensland, Queensland, Australia, 2003.
- [62] Mike Brooks, “VoiceBox: Speech Processing Toolbox for MATLAB”, disponível em <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>, consultado em Julho de 2009.
- [63] Free Software Foundation, “GNU Public License”, <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/copying.txt>, consultado em Julho de 2009.
- [64] Lastfm, “Music community website”, [www.lastfm.com](http://www.lastfm.com), consultado em Julho de 2009.
- [65] A. Sorensen, A. Brown, “jMusic - Music Composition in Java”, disponível em <http://jmusic.ci.qut.edu.au/>, consultado em Julho de 2009.



- [66] MathWorks, “MATLAB – The Language Of Technical Computing”, disponível em <http://www.mathworks.com/products/matlab/>, consultado em Julho de 2009.
- [67] Sun Microsystems, “Java”, disponível em <http://java.sun.com/>, consultado em Julho de 2009.
- [68] Free Software Foundation, “GNU Public License”, <http://www.gnu.org/copyleft/gpl.html>, consultado em Julho de 2009.
- [69] ESAC, “Essen Associative Code and Folksong Database”, disponível em <http://www.esac-data.org/>, consultado em Julho de 2009.
- [70] Eleanor Selfridge-Field, “*Beyond MIDI: the handbook of musical codes*”, MIT Press, ISBN 0262193949, 1997, pages: 343 – 361.
- [71] MIREX, “MIREX - Symbolic Melodic Similarity”, disponível em [http://www.music-ir.org/mirex/2007/index.php/Symbolic\\_Melodic\\_Similarity](http://www.music-ir.org/mirex/2007/index.php/Symbolic_Melodic_Similarity), consultado em Julho de 2009.
- [72] Carlos Esteban Gómez Chacón, “Algoritmos de Similitud Melódica para Recuperación de Información Musical”, Proyecto de Grado, Universidad Simón Bolívar, Sartenejas, 2007.
- [73] Object Management Group, “Unified Modeling Language”, disponível em <http://www.uml.org/>, consultado em Julho de 2009.
- [74] F. Barbosa, "Similarity-based Retrieval in High Dimensional Data with Recursive Lists of Clusters: A Study Case with Natural Language Dictionaries," icime, pp.432-436, 2009 International Conference on Information Management and Engineering, 2009.
- [75] F. Barbosa, F. Costa, “Timbre Similarity Search with Metric Data Structures”, aceite para publicação em Workshop on Exploring Musical Information Spaces (WEMIS) 2009.



## **Anexos**

### **A1 Parametrização das estruturas de dados métricas**

No estudo realizado sobre as várias estruturas de dados métricas (capítulos 5 e 6) é possível observar que estas apresentam parâmetros para a sua construção, com a excepção da estrutura VPTree. Com base nas referências bibliográficas de cada estrutura é observado que a escolha dos valores destes parâmetros afecta de forma significativa a performance das pesquisas por alcance de cada estrutura. Procurando realizar uma avaliação “justa” da pesquisa por alcance em cada estrutura de dados métrica, nesta secção é realizado um estudo sobre a parametrização de cada estrutura. Este estudo tem como objectivo, obter uma parametrização óptima para cada estrutura, de modo a efectuar a avaliação da pesquisa por alcance nas diferentes estruturas de dados métricas.

O método utilizado neste estudo da parametrização é semelhante ao descrito na secção 8.1, ou seja, para cada parametrização são construídas três instâncias da mesma estrutura de dados métrica, onde são realizadas diversas pesquisas com os elementos do conjunto de pesquisa, calculando o número médio de cálculos de distâncias em cada instância. O resultado final resulta na média dos valores médios obtidos em cada instância, observando a parametrização que minimiza este resultado final.

Neste estudo, apenas é utilizado um valor para o raio de pesquisa, que corresponde a 60% da distância média de cada espaço métrico respectivamente. A parametrização das estruturas de dados depende do espaço métrico, deste modo para cada estrutura são testadas várias parametrizações para cada espaço métrico.

Nas secções seguintes são apresentados os resultados do estudo efectuado, para cada estrutura de dados métrica, com a excepção da VPTree, a qual não possui parâmetros.

#### **A1.1 Parametrização da LAESA**

A estrutura de dados métrica LAESA possibilita a parametrização do número de elementos da base de dados que actuam como pivôs.

Nos gráficos da figura 57 é visível o comportamento da LAESA conforme a variação do número de pivôs em cada espaço métrico.

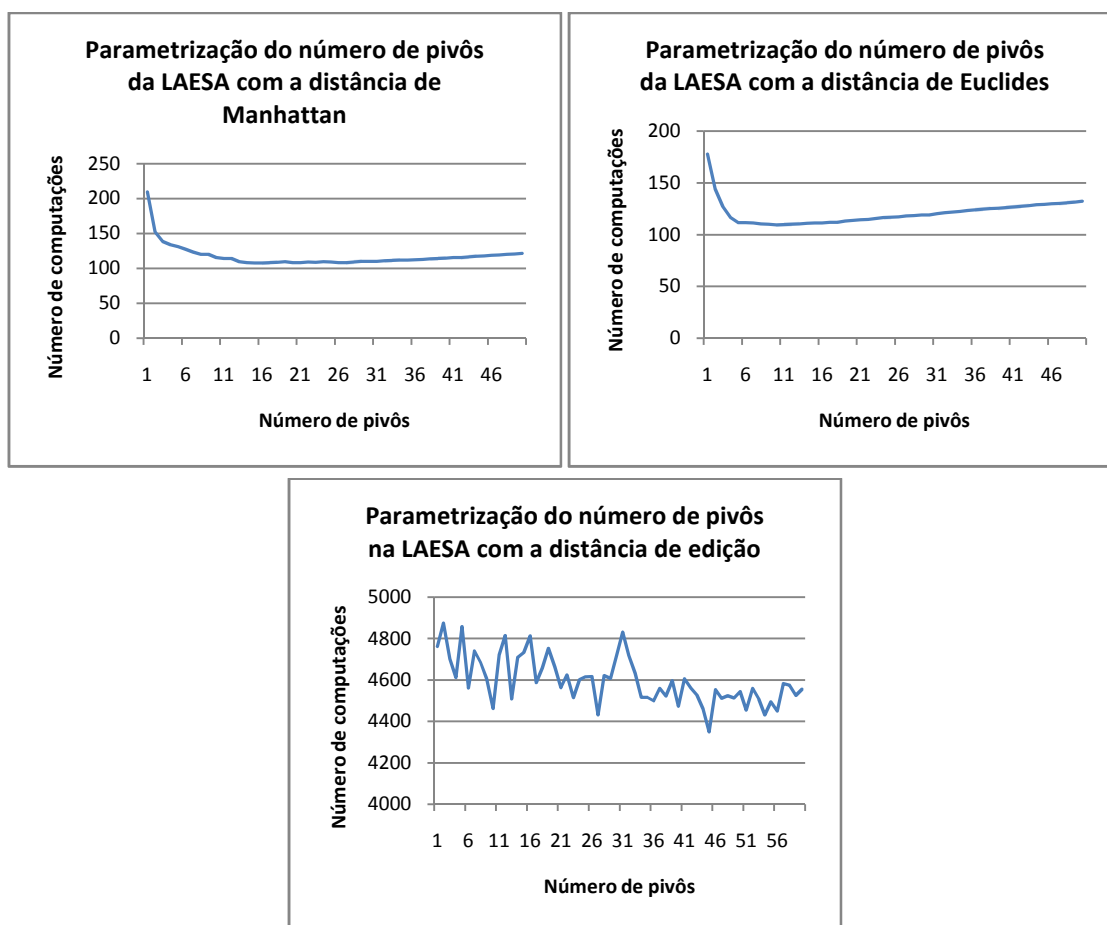
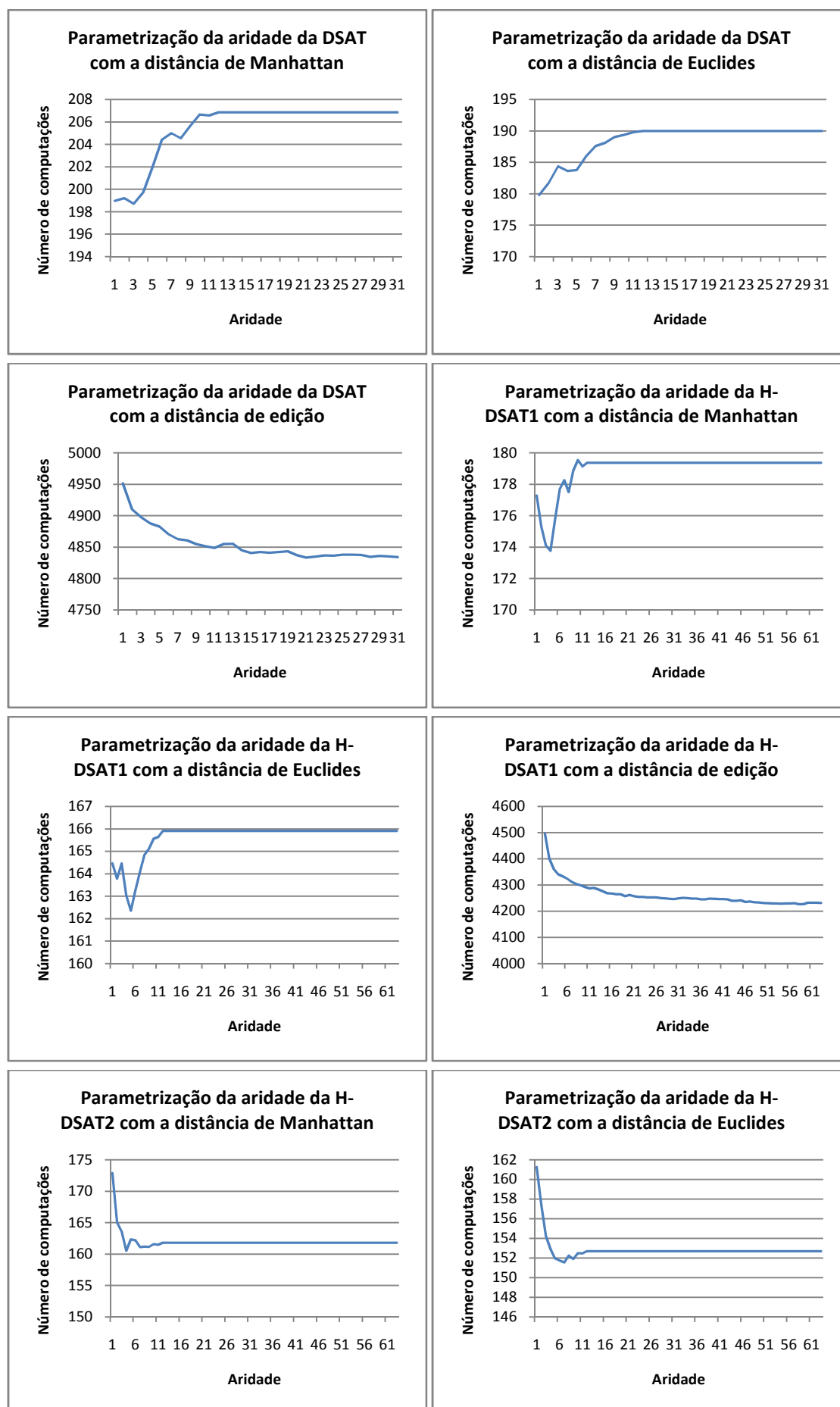


Figura 57. Parametrização da LAESA.

## A1.2 Parametrização da DSAT, H-DSAT1 e H-DSAT2

A estrutura de dados métrica DSAT e as suas variantes H-DSAT1 e H-DSAT2 possibilitam a parametrização do número de elementos da base de dados em cada nó, ou seja, a aridade da árvore.

Nos gráficos da figura 58 é visível o comportamento destas estruturas conforme a variação da aridade da árvore em cada espaço métrico.



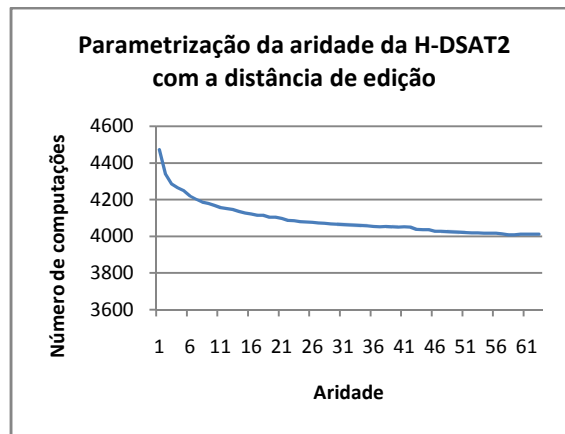
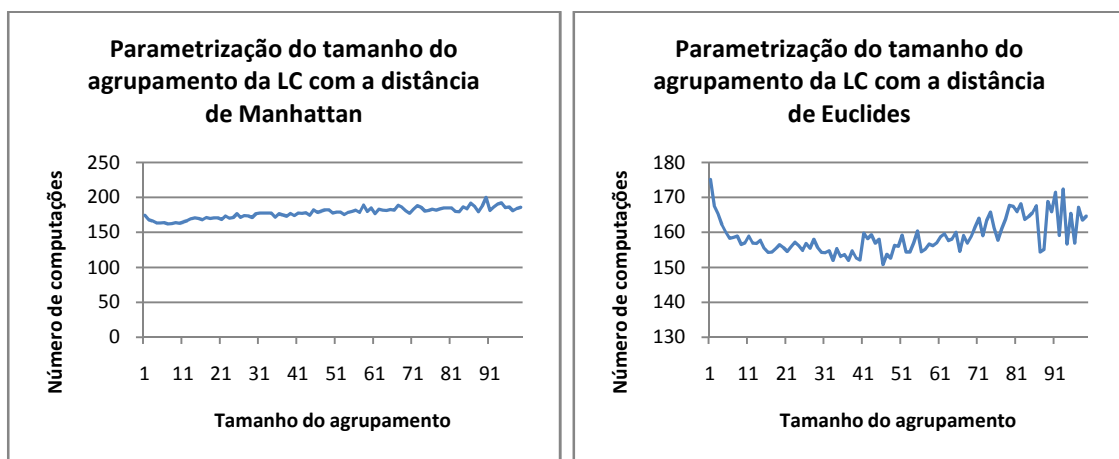


Figura 58. Parametrizações da DSAT, H-DSAT1 e H-DSAT2.

### A1.3 Parametrização da LC

A estrutura de dados métrica LC possibilita duas versões diferentes. Uma versão em que os agrupamentos possuem tamanho fixo e uma outra em que os agrupamentos possuem raio fixo. Na primeira é possível parametrizar o tamanho dos agrupamentos, enquanto na segunda é possível parametrizar o valor do raio dos agrupamentos.

Nos gráficos da figura 59 é visível o comportamento das duas versões da LC, conforme a variação do tamanho dos agrupamentos e a variação do raio dos agrupamentos, para a primeira e segunda versão respectivamente.



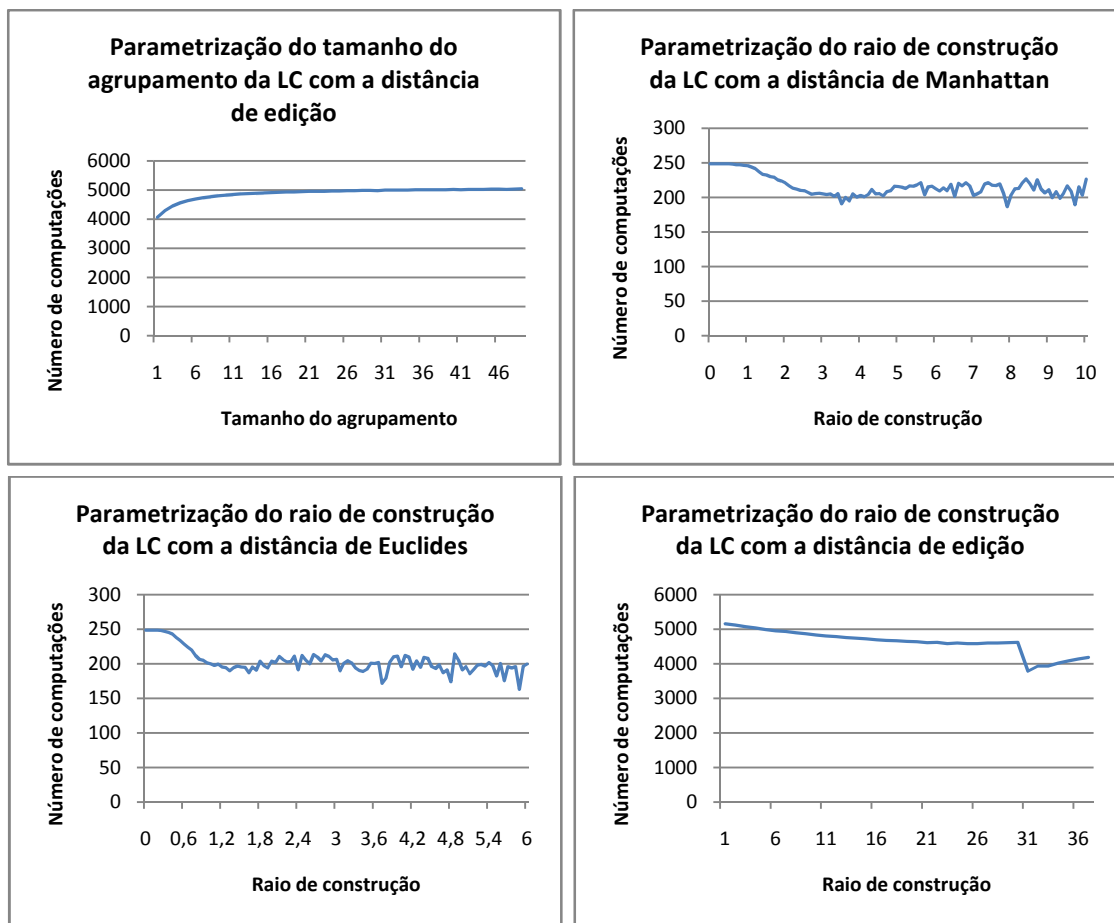
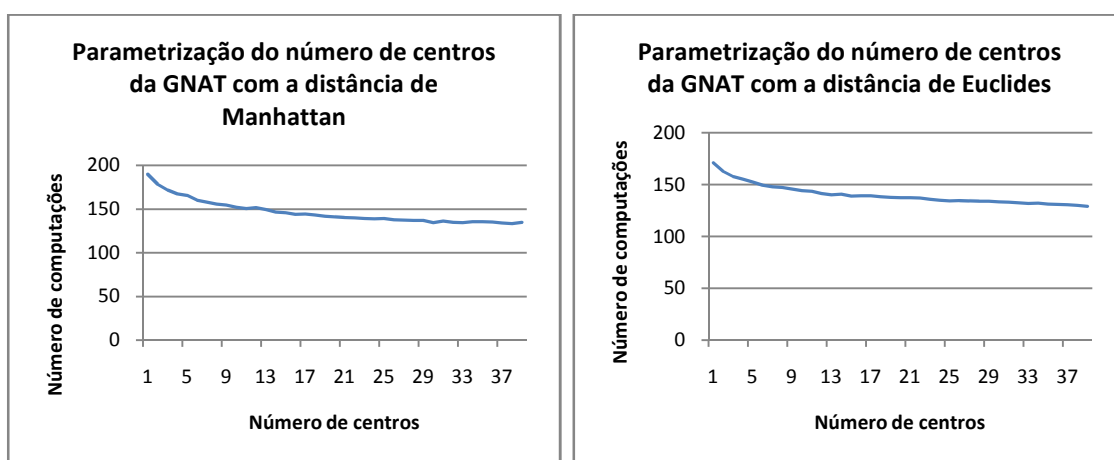


Figura 59. Parametrizações das duas versões da LC.

#### A1.4 Parametrização da GNAT

A estrutura de dados métrica GNAT possibilita a parametrização do número de elementos da base de dados em cada nó que actuam como centros (“degree”).

Nos gráficos da figura 60 é visível o comportamento da GNAT conforme a variação do número de centros (“degree”) em cada espaço métrico.



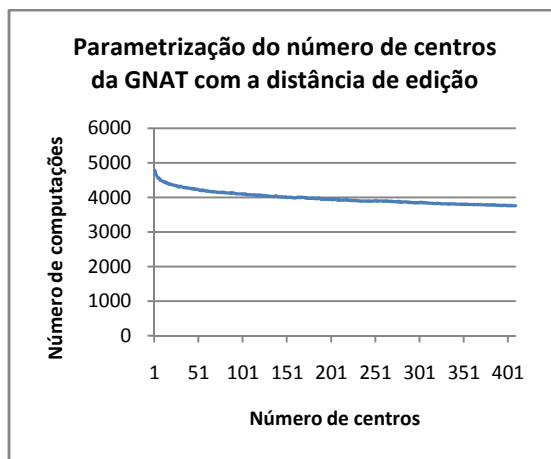
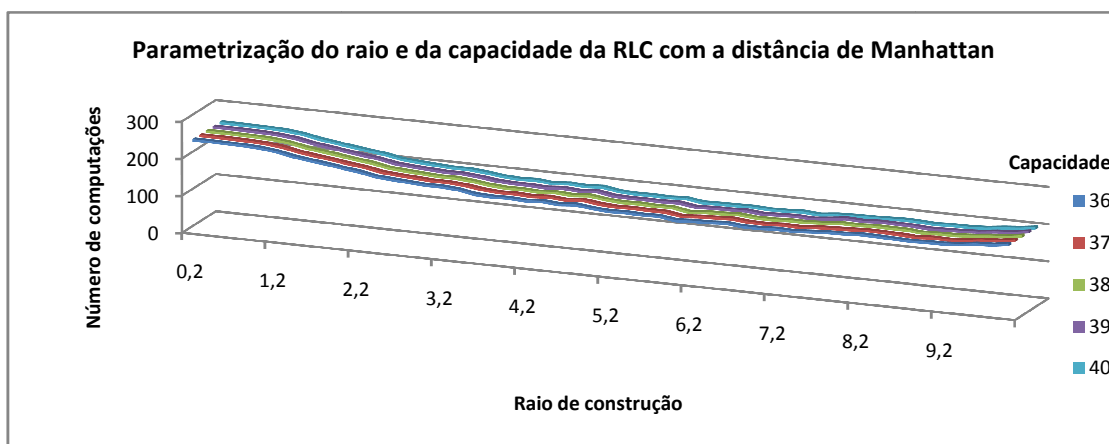


Figura 60. Parametrizações da GNAT.

A parametrização desta estrutura para cada espaço métrico apresenta a particularidade de a performance melhorar sempre que se aumenta o “degree”, até ao valor limite igual ao número de elementos de cada espaço métrico. No entanto, nesta situação a estrutura teria apenas um único nível e a tabela das distâncias de cada centro teria uma grande dimensão. Deste modo, os valores do “degree”, para cada espaço métrico, foram escolhidos quando um incremento deste valor resulta num pequeno acréscimo de performance.

### A1.5 Parametrização da RLC

Nos gráficos da figura 61 é visível o comportamento da RLC conforme a variação do valor do raio de construção e da capacidade da RLC para cada espaço métrico.





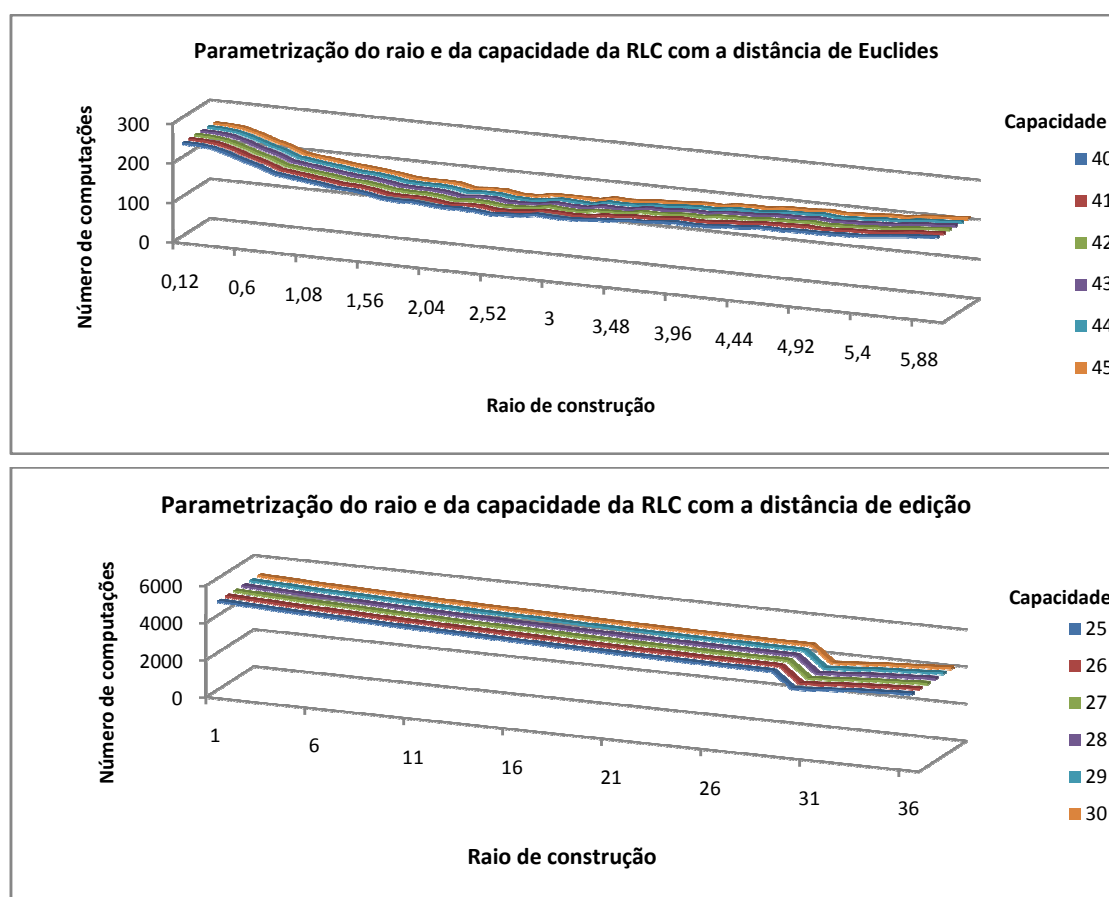


Figura 61. Parametrizações da RLC.

Nos gráficos não está presente a parametrização para outros valores da capacidade por motivos de legibilidade, no entanto foram testados os intervalos [20,45] e [15,30], para os espaços métricos da dimensão timbre e para o espaço métrico da dimensão melodia, respectivamente. Com base nestas parametrizações é possível observar que o parâmetro da capacidade da RLC não altera a performance de forma significativa, sendo o valor do raio de construção o parâmetro preponderante.